University of Alberta

Direct Numerical Simulations of Particulate Flows using a Finite Element/Fictitious Domain Approach

by

Carolina Diaz-Goano

A thesis
submitted to the Faculty of Graduates Studies and Research
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

in

Chemical Engineering

Department of Chemical and Materials Engineering

Edmonton, Alberta
Spring 2003

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisisitons et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-82094-7
Our file Notre référence
ISBN: 0-612-82094-7

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

Canada

University of Alberta

Library Release Form

Name of Author: *Carolina Diaz-Goano*

Title of Thesis: *Direct Numerical Simulations of Particulate Flows using a Finite Element/Fictitious Domain Approach.*

Degree: *Doctor of Philosophy*

Year this Degree Granted: *2003*

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Carolina D. Goano

Signature

6512 95 Ave - Edmonton
AB - T6B 1A7 - Canada

April 15, 2003

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Direct Numerical Simulations of Particulate Flows using a Finite Element/Fictitious Domain Approach** submitted by **Carolina Diaz-Goano** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Chemical Engineering**.

Dr. K. Nandakumar (supervisor)

Dr. P. Minev (supervisor)

Dr. J. H. Masliyah

Dr. C. Lange

Dr. G. Tryggvason (External examiner)

Date: April 14, 2003

To my parents, to MPB, and to all my loved ones, for their support and unconditional love have given me the confidence to follow my dreams, and the courage to take the risks that lead to success.

A mis padres, a MPB, y a todos mis seres queridos, su apoyo y amor incondicional me dieron la confianza para perseguir mis sueños, y el valor para tomar los riesgos que conducen al éxito.

Abstract

This thesis presents a new finite element method based on a Lagrange multiplier/fictitious domain approach for direct simulation of three dimensional multiphase flow problems involving Navier-Stokes equations coupled with rigid body equations. The basic idea of this new approach is to consider a fluid and particle velocity fields that are defined in the entire domain of interest. The particle field is restricted to be equal to zero outside the particles (an L^2 field) and the fluid field is defined everywhere in the domain, including the particles. The two systems of equations are additionally constrained by the assumption that the two fields are equal in the domain occupied by the particles. This additional linear constraint is imposed using a global (defined in the entire domain) Lagrange multiplier. The physical meaning of this multiplier is of the interaction force between the particles and the fluid. Our implementation uses unstructured finite elements for the spatial discretization. The time discretization is performed using a time-splitting method. This allows for a separate treatment of the generalized Stokes problem, the convection terms, and the rigid body (no-slip) constraint. A collision detection mechanism prevents particles from penetrating each other or the walls. The linear solver used is a preconditioned conjugate gradient. The solver has been successfully parallelized and performance has been explored. The code was validated by comparing the results for a spherical particle in different physical settings to experimental data obtained in our laboratory. For these experiments the motion of steel and nylon spheres was recorded using a high speed camera. The im-

ages obtained were then analyzed and position and velocity information was obtained for each case. Results presented include one particle settling under the gravity in a fluid initially at rest, wall effects on the terminal velocity and the interaction between several spherical particles where kissing, drafting and tumbling occurs. Speedups obtained with the parallel code are also shown.

Acknowledgments

I wish to thank my supervisors Dr. K. Nandakumar and Dr. P. Minev, for their valuable ideas, advice and encouragement. I am grateful to them for the freedom I had to explore different roads and for making my Ph.D. studies a very enjoyable experience.

Many thanks to Eldon and Damian for helping me to obtain the experimental data I needed to validate my model.

I would also like to acknowledge NSERC, iCore, the Government of the Province of Alberta and the Faculty of Graduate Studies and Research for the financial support.

Finally, I wish to thank my parents Graciela and John for their love, encouragement and advice now and throughout my life.

Contents

1 Introduction 1
- 1.1 Research goal . . . 1
- 1.2 Technical background . . . 4
- 1.3 Thesis overview . . . 6

2 Basis for the multiphase algorithm: 3-D Navier-Stokes solver 8
- 2.1 Time discretization: Operator Splitting . . . 10
  - 2.1.1 Method of Characteristics . . . 13
  - 2.1.2 Projection method: Pressure-Correction algorithm . . . 15
- 2.2 Spatial discretization . . . 17
- 2.3 Navier-Stokes solver validation . . . 21
  - 2.3.1 Case 1: Backward facing step . . . 21
  - 2.3.2 Case 2: Lid-driven cavity flow . . . 26
- 2.4 Conclusions . . . 31

3 A Lagrange multiplier/fictitious domain formulation 32
- 3.1 Governing equations . . . 32
- 3.2 Weak formulation in the combined fluid-particle space . . . 34
- 3.3 Fictitious domain formulation . . . 36
- 3.4 Numerical Scheme . . . 38
  - 3.4.1 Finite Element Discretization . . . 38
  - 3.4.2 Time Discretization . . . 39
- 3.5 Code validation . . . 43
- 3.6 Conclusions . . . 52

4 A Fictitious domain approach using global Lagrange multipliers 55
- 4.1 Starting formulation . . . 55
- 4.2 Discretization procedure . . . 59
- 4.3 Conclusions . . . 63

5 Computational implementation 65
- 5.1 Collision strategy . . . 65
- 5.2 Element Searching Algorithm . . . 67

5.3 Conclusions 70

6 Parallel implementation and other computational considerations 71
6.1 Parallel linear solver 71
6.2 Other computational considerations 73
6.3 Conclusions 73

7 Experimental velocity measurement of settling spheres 74
7.1 Experimental setup 74
7.2 Particle releasing mechanisms 75
7.3 2D vs. 3D measurements 77
7.4 Conclusion 77

8 Numerical and experimental results: single particle modeling 79
8.1 One particle settling in a vertical channel 79
8.2 One particle settling at higher Reynolds number 85
8.3 One particle in a rotating cylinder 94
8.4 Wall effects on the terminal velocity 99
8.5 Collision with walls 110

9 Numerical and experimental results: multiple particle modeling 123
9.1 Two sphere settling in horizontal configuration 123
9.2 Two sphere settling in vertical configuration 135
9.3 Particle-Particle Interactions: Drafting kissing and tumbling 140
9.4 Three spheres in an horizontal arrangement 155
9.5 Three spheres in a vertical arrangement 166
9.6 27 spheres settling 166
9.7 Performance of the parallel code 182

10 Conclusions 184

Appendix A. Numerical integration 193
A.1 Adapted Gaussian quadrature 193
A.1.1 One dimensional Gauss quadrature 193
A.1.2 Multi-dimensional Gauss quadrature 193

List of Tables

7.1 Physical properties of the fluids 74

8.1 Wall effects: initial position, horizontal distance traveled and terminal velocity. 99

9.1 Three spheres settling. Experimental setup and terminal velocity. . . 155
9.2 Number of processors, average time per pass of the conjugate gradient and speedup 183
9.3 Comparison of mesh size (number of nodes) with time spent in CG and speedups achieved 183

List of Figures

2.1 Finite element discretization: Taylor-Hood tetrahedron. 19
2.2 Test case: Backward facing step. 22
2.3 Backward facing step. Re=100, t=0.3 and 0.5 (dimensionless). 23
2.4 Backward facing step. Re=100, t=1.0 (dimensionless). 24
2.5 Backward facing step. Re=1, 50 and 100, t=0.1 (dimensionless). . . . 25
2.6 Lid-driven cavity test case. 27
2.7 Lid-driven cavity test case. a) Domain b) Re=100, t=0.01 (dimensionless). 28
2.8 Lid-driven cavity test case. Re=100, t=1.5 (dimensionless). 29
2.9 Lid-driven cavity. Comparison of velocity results (dimensionless) along the centerline of the cavity for Re=100. 30

3.1 Solid particles in a fluid domain. 33
3.2 Computational domain for one sphere in a channel. 45
3.3 Computational domain for one sphere. 46
3.4 Sphere's position at different times. Re=21.0 Fr=2.24. 47
3.5 Sphere settling in a channel, streamlines. 48
3.6 Sphere settling in a channel, relative motion. 49
3.7 Particle's vertical velocity vs. time (dimensionless). 50
3.8 Streamlines for Re=183.0 Fr=3.75. 53
3.9 Streamlines for Re=183.0 Fr=3.75. Wake formation. 54

5.1 Two particles near collision. 66

6.1 Conjugate gradient algorithm for solving $\mathbf{Ax} = \mathbf{b}$. 72

7.1 Experimental setup. 76
7.2 Analysis of data captured with a high speed camera. 76
7.3 Experimental setup: 3D information retrieval. 78

8.1 One particle settling. Experimental vertical velocity vs. time. Re=28.7. 80
8.2 One particle settling. Comparison of experimental and numerical vertical velocity vs. time (dimensionless). Re=28.7. 81
8.3 One particle settling. Experimental vertical velocity vs. time. Re=6.08. 82

8.4 One particle settling. Experimental and numerical vertical velocity vs. time (dimensionless). Re=6.0. 83
8.5 One particle falling under gravity. Re=41. 84
8.6 Particle falling under gravity. Re=50, Fr=1.12. Velocity vs. time (dimensionless). 86
8.7 One particle settling under gravity. Re=118.0, Fr=0.01. Streamtraces. 88
8.8 One particle settling under gravity. Re=118.0, Fr=0.01. Wake behind the sphere. 89
8.9 One particle settling. Re=220.0, Fr=0.01. Uy contour at x=0 (dimensionless). 90
8.10 One particle settling. Re=220.0, Fr=0.01. Streamtraces. 91
8.11 One particle settling. Re=300.0, Fr=0.01. Streamtraces. 92
8.12 One particle settling. Re=300.0, Fr=0.01. Streamtraces. 93
8.13 Sphere in a rotating cylinder: Angular velocity vs. timestep (dimensionless). 94
8.14 Sphere in a rotating cylinder, computational domain. 95
8.15 Sphere in a rotating cylinder, t=0.1 (dimensionless). 96
8.16 Sphere in a rotating cylinder, t=1.2 (dimensionless). 97
8.17 Sphere in a rotating cylinder, t=2.4 (dimensionless). 98
8.18 Wall effects: experimental results. 100
8.19 Wall effects: experimental results. Vertical velocity vs. time. 101
8.20 Wall effects: experimental results. Vertical velocity vs. time (dimensionless). 102
8.21 Wall effects: experimental results. Horizontal displacement vs. time. . 103
8.22 Wall effects: experimental results. Horizontal displacement vs. time (dimensionless). 104
8.23 Wall effects: numerical results. Case 1- Initial distance to the wall=0.1 (dimensionless). t=1dt and 150dt. 106
8.24 Wall effects: numerical results. Case 1- Initial distance to the wall=0.1 (dimensionless). t=350dt and 550dt. 107
8.25 Wall effects: numerical results. Case 1- Initial distance to the wall=0.1 (dimensionless). t=750dt and 950dt. 108
8.26 Wall effects: numerical results. Case 1- Initial distance to the wall=0.1 (dimensionless). t=1150dt and 1550dt. 109
8.27 Numerical results. Vertical velocity vs. time (dimensionless). Particle released at 0.1 (dimensionless) from the wall. 110
8.28 Numerical results. Horizontal displacement vs. time (dimensionless). Particle released at 0.1 (dimensionless) from the wall. 111
8.29 Numerical results. Displacement in Z vs. time (dimensionless). Particle released 0.1 (dimensionless) from the wall. 112
8.30 Numerical results. Vertical velocity vs. time (dimensionless). Particle released 0.3 (dimensionless) from the wall. 113

8.31 Numerical results. Horizontal displacement vs. time (dimensionless). Particle released 0.3 (dimensionless) from the wall. 114
8.32 Numerical results. Displacement is Z vs. time (dimensionless). Particle released 0.3 (dimensionless) from the wall. 115
8.33 Wall effects: numerical results. Streamtraces for t=500dt. 116
8.34 Wall effects: numerical results. Streamtraces for t=1000dt. 117
8.35 Wall collision: experimental results. t=0.04s, 0.08s, 0.10s, 0.108s, 0.116s and 0.12s. 118
8.36 Wall collision: experimental results. t=0.124s, 0.132s, 0.14s and 0.148s. 119
8.37 Wall collision: experimental results. t=0.156s, 0.164s, 0.20s, 0.24s. . . 120
8.38 Wall collision: Vertical velocity vs. time for a steel bearing. 121
8.39 Wall collision: Vertical velocity vs. time for a nylon sphere. 122

9.1 Two spheres in horizontal configuration. t=0s, 0.08s, 0.12s. 124
9.2 Two spheres in horizontal configuration. t=0.16s, 0.24s, 0.32s. 125
9.3 Two spheres in horizontal configuration. t=0.40s, 0.48s, 0.56s. 126
9.4 Two spheres in horizontal configuration. Experimental results: Vertical velocity vs. time. 127
9.5 Two spheres in horizontal configuration. Numerical results at t=0.01 (dimensionless). 128
9.6 Two spheres in horizontal configuration. Numerical results at t=2.0 (dimensionless). 129
9.7 Two spheres in horizontal configuration. Numerical results at t=4.0 (dimensionless). 130
9.8 Two spheres in horizontal configuration. Numerical results at t=6.0 (dimensionless). 131
9.9 Two spheres in horizontal configuration. Numerical results at t=8.0 (dimensionless). 132
9.10 Two spheres in horizontal configuration. Numerical results at t=11.0 (dimensionless). 133
9.11 Two spheres in horizontal configuration. Experimental and numerical results: Vertical velocity vs. time (dimensionless). 134
9.12 Two spheres in horizontal configuration. Numerical results: Z vs. time (dimensionless). 134
9.13 Two spheres in vertical configuration. Numerical results at t=0.01 (dimensionless). 136
9.14 Two spheres in vertical configuration. Numerical results at t=2.0 (dimensionless). 137
9.15 Two spheres in vertical configuration. Numerical results at t=4.0 (dimensionless). 138
9.16 Two spheres in vertical configuration. Numerical results at t=5.7 (dimensionless). 139

9.17 Two spheres kissing, drafting and tumbling. t=0.584s, 0.664s, 0.744 and 0.824s. 141
9.18 Two spheres kissing, drafting and tumbling. t=0.904, 0.984s, 1.064s and 1.224s. 142
9.19 Two spheres kissing, drafting and tumbling. 3D - t=0s, 0.04s and 0.08s. 143
9.20 Two spheres kissing, drafting and tumbling. 3D - t=0.12s, 0.16s and 0.2s. 144
9.21 Two spheres kissing, drafting and tumbling. 3D - t=0.24s, 0.28s and 0.32s. 145
9.22 Two spheres kissing, drafting and tumbling. 3D - t=0.36s. 146
9.23 Two spheres kissing, drafting and tumbling. Experimental results. Vertical velocity vs. time (dimensionless). 146
9.24 Two spheres kissing, drafting and tumbling: Numerical results for t=0.01 (dimensionless). Re=108. 148
9.25 Two spheres kissing, drafting and tumbling: Numerical results for t=0.20 (dimensionless). Re=108. 149
9.26 Two spheres kissing, drafting and tumbling: Numerical results for t=0.40 (dimensionless). Re=108. 150
9.27 Two spheres kissing, drafting and tumbling: Numerical results for t=0.60 (dimensionless). Re=108. 151
9.28 Two spheres kissing, drafting and tumbling: Numerical results for t=0.80 (dimensionless). Re=108. 152
9.29 Two spheres kissing, drafting and tumbling: Numerical results for t=1.0 (dimensionless). Re=108. 153
9.30 Two spheres kissing, drafting and tumbling: Numerical results for t=1.15 (dimensionless). Re=108. 154
9.31 Three spheres settling. Experimental results for spheres released at the same vertical height. t=0.20s, 0.60s, 0.80s and 1.0s. 156
9.32 Three spheres settling. Experimental results. Terminal velocity vs. time. 157
9.33 Three spheres settling. Numerical results. t=1dt (dimensionless). . . 158
9.34 Three spheres settling. Numerical results. t=150dt (dimensionless). . 159
9.35 Three spheres settling. Numerical results. t=350dt (dimensionless). . 160
9.36 Three spheres settling. Numerical results. t=550dt (dimensionless). . 161
9.37 Three spheres settling. Numerical results. t=750dt (dimensionless). . 162
9.38 Three spheres settling. Experimental and numerical results. Terminal velocity vs. time (dimensionless). 163
9.39 Three spheres settling. Numerical results. X displacement vs. time (dimensionless). 164
9.40 Three spheres settling. Numerical results. Z displacement vs. time (dimensionless). 165
9.41 Three spheres in a vertical arrangement. Experimental results: positions for t=0.04s, 0.08s, 0.16s and 0.32s. 167

9.42 Three spheres in a vertical arrangement. Experimental results: vertical velocity vs. time. 168
9.43 Three spheres in a vertical arrangement. Numerical results: t=1dt (dimensionless). 169
9.44 Three spheres in a vertical arrangement. Numerical results: t=15dt (dimensionless). 170
9.45 Three spheres in a vertical arrangement. Numerical results: t=25dt (dimensionless). 171
9.46 Three spheres in a vertical arrangement. Numerical results: t=35dt (dimensionless). 172
9.47 Three spheres in a vertical arrangement. Numerical results: t=55dt (dimensionless). 173
9.48 Three spheres in a vertical arrangement. Numerical results: t=75dt (dimensionless). 174
9.49 Three spheres in a vertical arrangement. Numerical results: t=95dt. Contour at X=0.12 (dimensionless). 175
9.50 Three spheres in a vertical arrangement. Numerical results: t=95dt (dimensionless). Side view. 176
9.51 Three spheres in a vertical arrangement. Numerical results: t=95dt. Contour at X=-0.10 (dimensionless). 177
9.52 Settling of 27 spheres. Computational domain. 178
9.53 Settling of 27 spheres. t=1.00 (dimensionless). 179
9.54 Settling of 27 spheres. t=3.00 (dimensionless). 180
9.55 Settling of 27 spheres. t=4.00 (dimensionless). 181
9.56 Time and Speedup vs. number of processors 182

Nomenclature

β	Small real number
β_m	Correction factor at iteration level m
$\boldsymbol{\lambda}$	Lagrange multiplier
$\boldsymbol{\omega}_i$	Angular velocity of the ith solid particle
$\boldsymbol{\phi}$	Quadric basis functions
$\mathbf{r}_i$	Radius of the ith particle
$\boldsymbol{\sigma}$	Stress tensor
$\mathbf{U}_i$	Velocity of the center of mass of the ith solid particle
$\Delta\mathbf{x}$	Spatial step
Δt	Time step
Δx	Minimum inter-nodal spacing among the elements containing P_i
ℓ	Inner product for the weak formulation, body forces term
$\frac{D}{Dt}$	Material derivative
Γ	Boundary of domain
γ_m	Descent direction at iteration level m
$\mathbb{D}$	Discretized domain space
θ_i	Angular position (spherical coordinates vector) of the ith solid particle
$\mathbf{D}[\,]$	Rate of strain tensor
$\mathbf{f}$	Force acting on a unit volume of material
$\mathbf{F}_i$	Force acting on the ith solid particle

$\mathbf{g}$	Acceleration of gravity
$\mathbf{I}$	Identity matrix
$\mathbf{I}_i$	Inertia tensor of the ith solid particle
$\mathbf{L}$	Gradient matrix
$\mathbf{M}$	Mass matrix
$\mathbf{N}$	Advection matrix
$\mathbf{n}$	Outward normal
$\mathbf{S}$	Stiffness or Laplacian matrix
$\mathbf{T}_i$	Torque acting on the ith solid particle
$\mathbf{u}$	Velocity of the fluid
$\mathbf{u}^*$	Intermediate velocity
$\mathbf{u}^n$	Velocity at time level n
$\mathbf{u}_0$	Initial velocity
$\mathbf{u}_n$	Normal component of the velocity
$\mathbf{u}_\tau$	Tangential component of the velocity
$\mathbf{V}$	Velocity space
$\mathbf{w}$	Weighting functions
$\mathbf{X}$	Position vector
$\mathbf{X}_i$	Position (Cartesian vector) of the ith solid particle
$\mathbf{Z}^h$	Functional space for mesh spacing h
$\mathcal{C}$	Convective operator
$\mathcal{D}$	Diffusion operator
$\mathcal{I}$	Identity operator
$\mathcal{L}$	Inner product for the weak formulation, gradient term
$\mathcal{M}$	Inner product for the weak formulation, mass term

$\mathcal{N}$	Inner product for the weak formulation, convection term
$\mathcal{Q}$	Integrating factor
$\mathcal{S}$	Inner product for the weak formulation, diffusion term
Cu	Courant number
sup	Supreme
μ	Viscosity
$\nabla\cdot$	Divergence
∇	Nabla - gradient operator
ν	Kinematic viscosity
Ω	Open bounded region in $\mathbf{R}^d$
$\partial\Omega$	Boundary of Ω
ψ	Linear basis functions
$\mathbf{R}^d$	d dimensional Euclidean space
ρ_F	Density of the fluid
τ_0	Time factor for the current level
τ_1	Time factor for the previous level
τ_2	Time factor for two levels before
τ_{95}	Time it takes the particle to reach 95% of the limit speed
$\tilde{\mathbf{u}}$	Convected velocity
$\widehat{\mathbf{V}}$	Velocity test function
$\widehat{\mathbf{v}}$	Velocity test function
$\widehat{\xi}$	Angular velocity test function
d	Space step in the discretized domain
D_c	Hydraulic diameter
G	Unstructured grid

H	Hilbert space
h	Mesh spacing
H^1	Hilbert space of square-Lebesgue-integrable functions
H_0^1	Hilbert space of continuous square-Lebesgue-integrable functions
L^2	Space of square-Lebesgue-integrable functions
$L^i_{k,l}$	k, l entry of the gradient matrix in the i direction
M_i	Mass of the ith particle
M_s	Mass of the solid particle
$M_{i,j}$	i, j entry of the mass matrix
N_g	Total number of points in the unstructured mesh
N_x	Number of voxels in the x direction
N_y	Number of voxels in the y direction
N_z	Number of voxels in the z direction
$N_{k,l}$	k, l entry of the advection matrix
N_p	Pressure degrees of freedom
N_v	Velocity degrees of freedom
p	Pressure
p^*	Intermediate pressure
p^n	Pressure at time level n
P_m	Point in the unstructured mesh
Q	Functional space
q	Weighting functions
s	Time variable
$S_{i,j}$	i, j entry of the stiffness or Laplacian matrix
t	Time

u_x Velocity in the x direction

u_y Velocity in the y direction

u_z Velocity in the z direction

u_{average} Mean or average velocity

u_{max} Maximum velocity

U_{P_i} Velocity at point P_i in the unstructured grid

V_n Voxel in the structured grid

v_{exp} Experimental sedimentation velocity

V_{Terminal} Terminal velocity of a particle settling

x Position vector of a mesh point

Fr Froude Number

Re Reynolds number

Chapter 1

Introduction

1.1 Research goal

Particulate flow exists in numerous engineering applications. During the last years numerical simulations have gained popularity since they provide a way to increase the understanding of multi-phase problems. The motion of particles through a fluid under the influence of gravity and the interaction between these particles is of great importance in areas such as sedimenting and fluidized suspensions, slurry transport and hydraulic fracturing.

In fluidized beds, a liquid or gas is passed upstream so that weight and drag forces are balanced. Fluidized beds are designed to enhance heat transport and promote chemical reactions. Example applications are drying processes, coal combustion and catalyst cracking for the conversion of light crude oils.

Slurry transport is another important application. Solid-liquid mixtures in pipes tend to segregate into a high solid concentration region in the center of the pipe surrounded by liquid with low concentration of solids near the walls of the pipe walls giving rise to lubrication. The problem here is to determine the nature of the forces which push particles away from the wall and to examine the different ways in which the holdup of the solids may develop [32]. Coal slurries in water, and water lubricated transport of heavy crude are well known examples of lubricated transport. A particular case of slurry transport is the hydro-transport in the bitumen extraction process of the oil sands. Oil sand and water are mixed together to make a slurry that is transported via pipeline. While the mixture of oil sand and water flows through the pipeline, large lumps of oil sand are broken down and bitumen is separated from

the oil sand in the form of tiny droplets, a process called oil sand conditioning. Oil sand hydro-transport is a much less expensive way to move the oil sand than the old method of using long conveyors. It is also a more flexible process and pipelines do not have to run in straight lines or over level ground. The other important benefit is that oil sand can be processed at lower temperatures, thus lowering energy consumption [65]. There are still some important questions to answer regarding the oil sand hydro-transport technology. Some of the major challenges include the effects of fine solids (clays), temperature, bitumen content (oil sand grade), and average sand grain size on the conditioning process and on pipeline friction losses. Effects of transporting large rocks on the pipeline pumps and the pipeline itself have yet to be determined [65].

Hydraulic fracturing, is a technique that allows oil or natural gas to move more freely from the rock pores where they are trapped to a producing well that can bring the oil or gas to the surface. After a well is drilled into a reservoir rock that contains oil, natural gas, and water, every effort is made to maximize the production of oil and gas. One way to improve or maximize the flow of fluids to the well is to connect many pre-existing fractures and flow pathways in the reservoir rock with a larger fracture. This larger, man-made fracture starts at the well and extends out into the reservoir rock for as much as several hundred feet. The man-made or hydraulic fracture is formed when a fluid is pumped down the well at high pressures for short periods of time (hours). The high pressure fluid (usually water with some specialty high viscosity fluid additives) exceeds the rock strength and opens a fracture in the rock. A propping agent, usually sand carried by the high viscosity additives, is pumped into the fractures to keep them from closing when the pumping pressure is released. The high viscosity fluid becomes a lower viscosity fluid after a short period of time. Both the injected water and the now low viscosity fluids travel back through the man-made fracture to the well and up to the surface [66].

This results in high permeability pathways through the reservoir that increase the effective permeability of the rock and therefore, improved extraction of the fluids. Due to the high cost involved and the desire to improve the economy of hydrocarbon production, computer packages are used to design the most efficient hydraulic fractures possible. The idea is to initiate fractures in the most desirable directions and extend them only within the pay-zone thereby being as economical as possible [19].

One of the problems of this technique is that during pumping sand particles migrate to the center plane leaving a clear fluid layer. This layer lubricates the motion

of the slurry thus increasing the settling of the sand which accumulates at the bottom of the fracture, reducing the well productivity. It can also interfere with the fracture growth process by blocking downward extension. Modeling of these processes could increase the understanding of the phenomena that take place and this in turn could lead to great cost savings.

Numerical modeling and simulation of scientific and engineering problems continuously demand greater computational power than is currently available. Computer output has increased in the last years but so have the problem dimensions, accuracy of the results and speed requirements. As our problems become more complex, it takes increasingly more time to simulate them. This large and complex problems are known otherwise as Grand Challenges. Grand Challenge applications address computation-intensive fundamental problems in science and engineering whose solutions can be advanced by applying high performance computing and communications technologies and resources [28]. Grand Challenges involve translating a mathematical model of physical phenomena into a program that requires the computer to carry out calculations and finding an accurate solution could demand teraflop (10^{12} floating point operations per second) of computer performance and 100 gigabyte memories [13]. A Grand Challenge problem is one that cannot be solved in a reasonable amount of time with today's computer [76]. These Grand Challenges include climate modeling, analysis of fuel combustion, ocean modeling, modeling the universe, modeling of dynamical systems, chemical dynamics and economic modeling.

One way of increasing the computational speed, a way that has been considered for many years, is by using multiple processors operating together on a single problem. We have nearly reached our limit as to how fast a single processor can run, due to physical restrictions of the silicon used in the manufacture of computer chips as well as the speed of light. Faster speeds are being obtained using massively parallel machines as well as using very fast networks of machines. Many computer simulation problems are inherently parallel, especially the Grand Challenge problems. Combining the power of several high performance computers offers one path to achieving teraflop performance.

Taking into account the importance of particulate flow in industrial problems and considering that simulating such problems is an extremely computational intensive task we set the goal of this research:
The primary goal of this research is to explore and implement highly efficient parallel algorithms to simulate the motion of many solid particles in three dimensional

Newtonian fluid flows.

Doing so we want to:

- advance the understanding of multiphase flow;
- provide an alternative way to improve equipment design and process diagnosis;
- provide insight on the form of closure models used in volume averaged equations and give a basis for testing and tuning some of the existing averaged multiphase flow models.

1.2 Technical background

The mathematical description of multiphase flows, like the mathematical description of turbulence, is still a significant challenge for the mathematicians and engineers working in the numerical simulation area. The models based on the interpenetrating continua hypothesis suffer of the same closure problems as in the case of the Reynolds-averaged turbulence models. Therefore, in the recent years some efforts were concentrated on the direct simulation of multiphase flows which would allow, again similar to the turbulence case, for a more accurate solution of the closure problems. This is a very demanding computational task, which required the development on new discretization techniques and computer implementations.

Different approaches have been used to simulate multiphase flows. Traditionally, problems involving solid-fluid flows were solved using the continuum theory where both phases are viewed as a interpenetrating mixture (see Zang et al. [77]). In such an approach, volume average equations for the velocity and the particle concentration are solved. The interaction of transfer processes between phases is unknown and has to be represented by closure models.

Other simulation techniques that provide a better description of the multiphase flow are the Lagrangian particle tracking or Lagrangian numerical simulation (LNS) and direct numerical simulation (DNS). In the first case both the particle and fluid motion are described. The fluid flow is solved solving the Navier-Stokes equations in all the domain, including the volumes occupied by the particles. The particles are moved solving rigid body equations. The forces and torques that account for the interaction between solid and fluid are not computed from the calculated fluid field but described by empirical correlation and do not account for particle-particle interaction or particle-boundary interactions.

In DNS the hydrodynamic forces and torques are solved without approximation and computed directly from the fluid field. In these methods the fluid flow is governed by the continuity and momentum equations whereas the particles are governed by the equation of motion for a rigid body. Flow field around each individual particle is resolved; the hydrodynamic force between the particle and the fluid is obtained from the solution and is not modeled by any drag law.

Direct simulation of the motion of solid particles in fluids was started by Hu et al. [39] who simulated the motion of circular particles in sedimenting and shear flows at particle Reynolds numbers in the hundreds. Unverdi and Tryggvason introduced a front tracking/finite difference method for computing the unsteady motion of drops and bubbles [74]. In their work, the drop surface is tracked by separate computational points that are moved on a fixed grid by interpolating their velocity from the grid. These points form a front that is used to keep the density and viscosity stratification and to calculate surface tension forces. Their method has not been applied to solid-liquid flows.

Joseph et al. [26] used direct simulations for initial value problems of two-dimensional motion of circular and elliptical particles in sedimenting, Couette and Poiseuille flows of a Newtonian fluid at particle Reynolds numbers in the hundreds. Recently Hu et al. [40] conducted DNS of fluid-solid systems using an Arbitrary-Lagrangian-Eulerian (ALE) technique that uses a moving mesh scheme to handle the time-dependent fluid domain. A new mesh is generated when the old one becomes too distorted and the flow field is projected onto the new mesh.

Numerical simulations for flows of incompressible fluids containing rigid bodies can be carried out using a body fitted mesh, that needs to be updated at each time step to account for the body's movements. Tryggvason and collaborator [73] studied extensively this approach. Other examples of such techniques are the ones presented by Hu [41] and Johnson and Tezduyar [45]. Moving grids, remeshing and doing projections is a time consuming procedure and the remeshing algorithm can be very complex.

Another approach is to use a set of fixed Eulerian grids (a grid that would describe the solid particles and another that would represent the fluid field) of simpler shapes and provide a mechanism for them to interact. This approach is called embedding or fictitious domain method. Fictitious domain methods were discussed by Buzbee et al. [10] , Astrakhatsev [5], and Peskin [61] and developed further by Glowinski et al. [34], [32]. To apply the fictitious domain approach to the problem of particulate flow, the

fluid flow is computed as if the space occupied by the particles was filled with fluid. The rigid body constraints are imposed through a no-slip boundary condition on the particle's boundaries. In the approach developed by Glowinski, Lagrange multipliers are associated with the boundary conditions on the particle's boundaries. This allows a fixed grid to be used, eliminating the need for remeshing, a definite advantage in parallel implementations. Pan et al. [60] studied the fluidization of a large number of spheres. Peskin [61] and LeVeque [50] used non Lagrange multiplier based fictitious domain.

1.3 Thesis overview

In the current study we present a new Lagrange multiplier/fictitious domain based technique to carry out the direct numerical simulation of incompressible viscous flow described by Navier-Stokes equations coupled with rigid body constraints. In our view, a fictitious domain approach is an effective method to solve for these complex and computational intense problems. Therefore, in a preliminary study we used the fictitious domain approach of [33] to develop a formally second order (in space and time) finite element scheme [23]. In this study, our numerical solver used unstructured finite elements for the space discretization and operator splitting for the time discretization. The Lagrange multipliers were defined locally to enforce the rigid body constraints. Then we modified it, in an attempt to make it more efficient and as a result we devised a new scheme, still based on the fictitious domain idea. The numerical experiments showed an excellent performance of this scheme. Its key novelty is that it approximates both, the fluid velocity field and the Lagrange multiplier for imposition of the rigid body motion on the same fixed Eulerian grid. This allowed us to avoid the gridding of the rigid particles. Using a properly weighted H^1 norm and the fact that the Lagrange multipliers are expanded over the same basis as the two velocity fields we were able to reduce the work for computing the fluid/rigid particles velocities and enhance its stability (this was verified only experimentally).

The rest of this thesis is organized as follows: Chapter 2 presents the mathematical formulation of the equations that govern the fluid motion and introduces a Navier-Stokes solver as the basis for the actual algorithm. Spatial and time discretization techniques are discussed in detail. Chapter 3 presents the fictitious domain/Lagrange multiplier method and shows the results obtained carrying on simulations using this technique. Chapter 4 describes the new algorithm, defining the inner products to

be used to enforce the rigid body constraints and describing the space and time numerical approximation used. Chapter 5 discusses some implementation details such as the particle-particle collision strategy implemented and the search algorithm used to efficiently locate a given point in an unstructured grid. Chapter 6 describes the parallelization of the code, in particular the linear solver and mentions some other implementation details. Chapter 7 describes the experimental setup and discusses the problems we had to overcome to obtain the terminal velocity measurements for the settling of solid spheres. Chapter 8 presents the experimental and numerical results for simulations involving a single particle. Chapter 9 presents the experimental and numerical results for simulations involving multiple particles. At the end of this chapter the parallel linear solver performance is analyzed. Finally, conclusions are presented in Chapter 10.

Chapter 2

Basis for the multiphase algorithm: 3-D Navier-Stokes solver

When modeling multiphase flow problems a non-linear system of equations has to be solved. These equations include those that describe the behavior of the fluid and those that describe the behavior of the solid particles. The solid particle and fluid equations are coupled through hydrodynamics forces and torques. To correctly describe the multiphase problem, the global system has to be additionally constrained in order to account for this coupling and to enforce the rigid body constraints.

The equations that describe the fluid behavior are the Navier-Stokes equations. As a first approach to build a multiphase simulation algorithm the Navier-Stokes equations were discretized and a code developed to solve three-dimensional incompressible flow problems. This solver and the numerical techniques developed for it are described below.

Let $\Omega \in \mathbf{R}^d$ be and open bounded domain with boundary Γ (sufficiently smooth). The equations that describe the unsteady flow of a fluid are:

Momentum equation

$$\rho_F \frac{D\mathbf{u}}{Dt} = \rho_F \mathbf{g} + \nabla \cdot \boldsymbol{\sigma} \text{ in } \Omega(t) \tag{2.1}$$

Mass Conservation Equation

The continuity equation can be written as:

$$\frac{d\rho}{dt} + \rho(\nabla \cdot \mathbf{u}) = 0 \qquad \text{in } \Omega(t) \tag{2.2}$$

And for an incompressible fluid equation (2.2) can be simplified to:

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega(t) \tag{2.3}$$

where $\mathbf{u}$ is the velocity vector, ρ_F is the density of the fluid, $\frac{D\mathbf{u}}{Dt}$ is the material derivative $\frac{D\mathbf{u}}{Dt} = \frac{D\mathbf{u}}{Dt} + (\mathbf{u} \cdot \nabla)\mathbf{u}$ and $\boldsymbol{\sigma}$ is the stress tensor.

For a Newtonian fluid the stress tensor takes the form of: $\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathbf{D}[\mathbf{u}]$, where p is the pressure of the fluid, $\mathbf{I}$ is the identity matrix, μ is the fluid viscosity and $\mathbf{D}[\mathbf{u}]$ is the rate of strain tensor ($\mathbf{D}[\mathbf{u}] = 1/2[\nabla\mathbf{u} + \nabla\mathbf{u}^T]$). Equations (2.1) and (2.3) are solved subject to the initial condition:

$$\mathbf{u}(\mathbf{X}, t) = \mathbf{u}_0 \text{ in } \Omega \tag{2.4}$$

and the boundary conditions

$$\mathbf{u}(\mathbf{X}, t) = g(\mathbf{X}, t) \text{ on } \Gamma \tag{2.5}$$

Finding a numerical solution for the above equations (momentum and mass conservation) is not trivial due to the non-linear characteristic of the system, the incompressibility condition (equation (2.3)) and the coupling of terms [2]. For incompressible fluids the mass conservation equation reduces to the incompressibility constraint requiring that the velocity be divergence free instantaneously in the entire computational domain. This can be achieved by adjusting the pressure. The mathematical importance of the pressure in an incompressible flow lies in the theory of the saddle-point problems where it acts as a Lagrange multiplier that constrains the velocity to remain divergence free [27], [30].

There are many ways to solve the Navier-Stokes equations. Solving the equations directly in coupled form leads to a very large system and since the pressure variable does not appear in the continuity equation, the matrix associated with such a system contains zeros in the main diagonal. To overcome this problem partial pivoting is required, but this would destroy the banded structure of the discrete system. Solving the system directly would require considerable computational time and power. Therefore, it would be desirable to find a solution by a method that could decouple the treatment of the velocity and the pressure. Using discrete spaces is a common and flexible approach, but the divergence issue must be dealt with directly. Two well known methods that use such an approach are the *penalty function method* [18] and the *projection method* [21].

In the first method the continuity equation is modified to:

$$\epsilon p + \nabla \cdot \mathbf{u} = 0 \tag{2.6}$$

where ϵ is a small number (perturbation parameter). The pressure can then be eliminated from the momentum equation and this equation can be decoupled from the continuity one. This introduces finite compressibility and solutions should be checked in the limit of $\epsilon -> 0$. A disadvantage of this method is that the perturbation parameter must be chosen carefully, small enough for the approximation to be accurate. The resulting system for the penalty methods yields an ill-conditioned matrix. Normally the penalty method is used for steady Navier-Stokes problems since for the unsteady problems the contribution of the unsteady inertia forces makes the system matrix ill-conditioned [71]. Another way to solve the unsteady Navier-Stokes equations is using a projection method [36], [37]. In this kind of method the velocity and pressure are decoupled by taking the divergence of the momentum equation. Thus, two equations are obtained: a Poisson equation for the pressure and a general convection-diffusion equation. The pressure equation can be solved using a preconditioning technique and the convection-diffusion equation can be solved using operator splitting. A projection method was used in the developing of the Navier-Stokes solver for the current study. The following sections present this method in further detail.

2.1 Time discretization: Operator Splitting

Standard Galerkin approximations of convection-diffusion problems are often unstable due to the fact that the resulting system is no longer diagonally dominant. For spectral methods the eigenvalues of the diffusion systems are real and strictly negative and grow with $O(N^4)$, if $N \rightarrow \infty$ (N being the order of approximation) [12], whereas for the low order finite elements and finite differences methods the eigenvalues of the diffusion operator globally grow with the number of collocation points like $O(N^2)$. On the other hand, the eigenvalues of the convective system are complex and grow with $O(N^2)$ [55]. Therefore, the diffusion (or stiff) part requires an integration algorithm that is stable in the negative real axis. Since implicit algorithms generally have a significantly larger stability region than comparable explicit methods, a (semi) implicit method would be an obvious choice since it would be able to provide solutions that are accurate at slow scales and stable at the fast scales. Also, the diffusion matrix does not depend on time so a suitable iterative solver can be applied at each time step. An

implicit time integration method requires that an algebraic system of equations be solve for each time step. In the case of Navier-Stokes equations an implicit method would require a significant fraction of the total computational time be devoted to solving the convective part (since the system is non-linear and the velocity depends on time, the convective matrix would have to be re-calculated at each time step). To achieve reasonable performance of the numerical scheme an explicit method has to be used. Explicit methods are generally simpler to implement and provide solutions of high-order accuracy at all scales. However, their stabilities requirements force the maximum integration step to be of the order of the fastest (smallest) time scale. For a stiff problem the ratio between the fastest scale and the scale at which the process evolves may be very large. For the case of the Navier-Stokes equations the eigenvalues of the convection part grow with $O(N^2)$ so the restriction on the time step is not too severe. The eigenvalues for the diffusion part are proportional to the inverse of the Reynolds number. Therefore, if an explicit time integration was to be used the restriction on the time step would become unreasonable. For the reasons given above, it would be desirable to separate the problem into a diffusion and a convective subproblem and use different methods to solve for each part. An operator splitting technique [51] would allow us to use any combination of time integration schemes for the different operators of the Navier-Stokes equations. In the current study we apply an operator splitting technique as presented in [54] and [55]. In this way, we solve the Stokes and continuity equation separately from the convective terms, using an adequate time integration scheme for each operators.

Any advection-diffusion equation can be written as:

$$\frac{d\theta}{dt} = \mathcal{D}\theta + \mathcal{C}\theta + \mathbf{f} \tag{2.7}$$

where $\mathcal{D}$ and $\mathcal{C}$ are operators involving different time levels: $\mathcal{D} = (\nabla \cdot \mu\nabla)$ is the diffusion operator and $\mathcal{C} = -(\mathbf{u} \cdot \nabla)$ is the convection operator. Following the work of Maday et al. [51] equation (2.7) can be written in terms of an integration factor $\mathcal{C}$:

$$\frac{\partial}{\partial t}(\mathcal{Q}_{\mathcal{C}}^{(t^*,t)}\theta(t)) = \mathcal{Q}_{\mathcal{C}}^{(t^*,t)}(\mathcal{D}\theta + \mathbf{f}) \tag{2.8}$$

with t^* an arbitrary fixed time. The integration factor $\mathcal{Q}_{\mathcal{C}}^{(t^*,t)}$ is defined by the initial value problem:

$$\frac{\partial}{\partial t}\mathcal{Q}_{\mathcal{C}}^{(t^*,t)} = -\mathcal{Q}_{\mathcal{C}}^{(t^*,t)}\mathcal{C} \tag{2.9}$$

$$\mathcal{Q}_{\mathcal{C}}^{(t^*,*)} = \mathcal{I} \tag{2.10}$$

where $\mathcal{I}$ is the identity operator.

In our case, $\mathcal{C}$ represents a second order Lagrange explicit scheme which is used to solve the convective part of the equation, $\mathcal{D}$ corresponds to a second order backward scheme used to solve the diffusion part.

The splitting scheme is summarized as follows:

Step 1: Solve the convection problem

The convection problem is stated as an initial value problem:

$$\begin{cases} \frac{\partial \tilde{\theta}(s)}{\partial s} = \mathcal{C}(\tilde{\theta})(s),\ 0 < s < i\Delta t, \\ \tilde{\theta}(0) = \theta^{n+1-i} \end{cases} \tag{2.11}$$

It follows that:

$$\mathcal{Q}_{\mathcal{C}}^{t^{n+1},t^{n+1-i}}\theta^{n+1-i} = \tilde{\theta}(i\Delta t) \tag{2.12}$$

Equation (2.11) can be solved with a suitable explicit scheme, such as a Runge-Kutta method [53] or the method of the characteristics, using a time step Δs that does not need to be equal to that used for solving the diffusion part of the Navier-Stokes equation. In the present study we compute the convection part of the Navier-Stokes equation using the *method of characteristics* as described in [54].

Step 2: Solve the diffusion problem

Equation (2.8) is integrated using a time integration method that is stable in the negative real axis. The chosen integration method is a backward differences scheme. Taking $t^* = t^{n+1}$ and applying a second order backward differences scheme with time step $\Delta t = t^{n+1} - t^n$ to equation (2.8) we obtain the following semi-discrete system:

$$\frac{\frac{3}{2}\theta^{n+1} - 2\mathcal{Q}_{\mathcal{C}}^{t^{n+1},t^n}\theta^n + \frac{1}{2}\mathcal{Q}_{\mathcal{C}}^{t^{n+1},t^{n-1}}\theta^{n-1}}{\Delta t} = \mathcal{D}\theta^{n+1} + f^{n+1} \tag{2.13}$$

To integrate equation (2.8) the terms $\mathcal{Q}_{\mathcal{C}}^{(t^{n+1},t^{n+1-i})}\theta^{n+1-i} (i = 1, 2)$ have to be evaluated. The explicit construction of $\mathcal{Q}_{\mathcal{C}}^{(t^{n+1},t^{n+1-i})}$ is avoided by solving the initial value problem described in Step 1. This second order scheme is applied to the Stokes problem and solved using a *pressure-correction method.*

The details of the method of characteristics and the pressure correction method are presented in the next two sections.

2.1.1 Method of Characteristics

There are mainly two basic ways to implement a characteristics method to solve the Navier-Stokes equations in the context of finite elements methods. One approach is to solve the diffusion and convection parts of the equation together forming a unified Galerkin formulation [62] [14]. This approach has good conservation properties but requires integrations of the basis functions based on the Lagrangian mesh within the Eulerian grid using numerical quadrature that may reduce the stability of the algorithm. The second approach is to treat the convection part separately from the other parts of the Navier-Stokes equations. The simplest and computationally cheapest way to integrate the convective terms is to integrate them directly point-wise along the characteristic lines, avoiding in this way the Galerkin formulation for them [54]. This second approach is the one developed in the current study.

In both cases interpolation of the Eulerian mesh onto the nodal points associated with the Lagrangian elements is required. In the case of unstructured grids this could be a complex and time consuming task. The speed of the element searching algorithm would highly depend on the goodness of the initial guess, and this in turn will depend on the time step used in the algorithm.

The advection part of Navier-Stokes equations can be written as:

$$\begin{aligned} &\frac{\partial \tilde{\mathbf{u}}^{n-i}(s)}{\partial s} = -(\tilde{\mathbf{u}}^{n-i}(s) \cdot \nabla)\tilde{\mathbf{u}}^{n-i}(s), 0 \leq s \leq (i+1)\Delta t, i = 0, ..k \\ &\tilde{\mathbf{u}}^{n-i}(0) = \mathbf{u} \end{aligned} \tag{2.14}$$

where $k = 0$ for a first order scheme and 1 for a second order one.

The characteristic method can be applied to equation (2.14) and summarized as follows:

1.1) Perform a second order accurate extrapolation of the velocity field.

$$\mathbf{u}^{n+1} = 2\mathbf{u}^{n} - \mathbf{u}^{n-1} \tag{2.15}$$

1.2) For each point in the Eulerian mesh solve the boundary problem described by equation (2.16) and determine the foot of the characteristic.

$$\begin{aligned} &\frac{d\mathbf{X}_{\mathbf{x}}^{n+1}(t)}{dt} = \mathbf{u}(\mathbf{X}_{\mathbf{x}}^{n+1}(t), t) \text{ in } [t^{n-i}, t^{n+1}] \\ &\mathbf{X}_{\mathbf{x}}^{n+1}(t^{n+1}) = \mathbf{x} \end{aligned} \tag{2.16}$$

This can be achieved using a first order Euler explicit scheme as described by equation (2.17).

$$\mathbf{X}_{\mathbf{x}}^{n+1}(t^{n-i}) = \mathbf{X}_{\mathbf{x}}^{n+1}(t^{n+1}) - i\Delta t\mathbf{u}^{n+1} \quad i = 0, 1 \tag{2.17}$$

Here $\mathbf{u}(\mathbf{X}_{\mathbf{x}}^{n+1}(t), t)$ is the velocity field obeying Navier-Stokes equations, $\mathbf{x}$ is an arbitrary point within the solution where the characteristic terminates at time level t^{n+1}, $\mathbf{X}_{\mathbf{x}}^{n+1}$ is the characteristic curve ending at point $\mathbf{x}$.

Note: Boukir et al. [8] showed that although this scheme is only first order in time, the overall algorithm is second order accurate.

1.3) Determine the elements containing the feet of the characteristics within the Eulerian finite element grid $\mathbf{X}_{\mathbf{x}}^{n+1}(t^{n-i})$.

1.4) Interpolate the velocity at the feet of the characteristic using the finite element interpolant based on the Eulerian grid at time t^{n-i}, $i = 0, 1$. This yields the convected velocity field $\tilde{\mathbf{u}}^{n-i}$.

$$\tilde{\mathbf{u}}^{n-i}(\mathbf{x}) = \mathbf{u}^{n-i}(\mathbf{X}_{\mathbf{x}}^{n+1}(t^{n-i})),\ i = 0, 1 \tag{2.18}$$

In this algorithm it is assumed that the characteristics never cross the domain boundary Γ. Difficulty could arise in case of inflow boundaries. To handle this problem substepping could be performed to guarantee that the characteristics originating at the nodes in the vicinity of the inflow boundary remain within the computational domain Ω, but this could lead to a severe time step restriction. In the present algorithm, the characteristics can cross the boundary. If that is the case, the inflow velocity profile is prescribed at the feet of the characteristic. This is equivalent of imposing that

$$\begin{aligned} \frac{\partial \mathbf{u_n}}{\partial \mathbf{n}} &= 0 \\ \frac{\partial \mathbf{u_\tau}}{\partial \mathbf{n}} &= 0 \end{aligned} \tag{2.19}$$

where $\mathbf{u_n}$ and $\mathbf{u}_\tau$ are the normal and tangential components of the velocity $\mathbf{u}$ at the inflow and $\mathbf{n}$ is the local outward normal to the boundary. At outflow boundaries no conditions are required in the Lagrangian substep since they are imposed during the solution of the generalized Stokes problem in the form of

$$\begin{aligned} \frac{\partial \mathbf{u_n}}{\partial \mathbf{n}} &= 0 \\ \mathbf{u}_\tau &= 0 \\ p &= 0 \end{aligned} \tag{2.20}$$

In the case of moving boundaries with a high curvature mesh (such as a cylinder rotating, presented in a later chapter), the feet of the characteristic are found according

to the problem's geometry and the velocity is calculated according to the prescribed velocity.

2.1.2 Projection method: Pressure-Correction algorithm

Projection methods are splitting techniques that approximate the Navier-Stokes equations with an advection-diffusion system for the momentum balance and an elliptic Poisson equation for the pressure. Projection methods were first proposed by Chorin [16] and Temam [70].

There are basically two approaches to projections methods: velocity correction and pressure correction methods. In both cases the incompressibility constraint (pressure) and the diffusion are treated in different substeps ([24], [49], [46]). In the first case, the divergence free constraint is imposed on the intermediate velocity not on the velocity time at the new time level.

In the second case, using an initial approximation of the pressure, the momentum equations are solved to obtain an intermediate velocity ("prediction"). The pressure Poisson equation is then solved for a pressure correction which is used to correct the intermediate velocity field and enforce incompressibility. When Chorin [16] first introduced the pressure-correction method the intermediate velocity was calculated neglecting the pressure terms from the Navier-Stokes equations. Subtracting that intermediate velocity from the original momentum equation, taking the divergence on the results and neglecting the diffusion terms, he obtained a Poisson equation for the pressure that was used to correct the intermediate velocity. The procedure derived by Chorin was first order accurate in time for the velocity. Chorin's method was then improved by van Kan [75] by calculating the intermediate velocity using the pressure at the previous time level. Using a Crank-Nicolson time integration scheme his method was second order in time for the velocity.

As described by Gresho et al. [36] projection method approximations generally proceed as follows:
Given a divergence free velocity field that satisfies the boundary and initial conditions perform the following steps:

1. Guess (approximate in some way) the concomitant pressure gradient both at $t = 0$ and for $t > 0$.

2. Solve the momentum equations alone up to the projection time $t = T$. This time could be either set a priori or defined as the time at which an appropriate norm of

the divergence of the resulting intermediate velocity reaches some predetermined maximum allowable value.

3. Correct the velocity performing the projection of the intermediate velocity onto the appropriate subspace of divergence free vector fields. This completes one projection cycle. Reset time and go to Step 1.

The system that results from equation (2.13) after applying the method of characteristics to obtain the convected velocity $\tilde{\mathbf{u}}$ is:

$$\begin{aligned} &\tfrac{3}{2\Delta t}\mathbf{u}^{n+1} - \mathcal{D}\mathbf{u}^{n+1} = \tfrac{2}{\Delta t}\tilde{\mathbf{u}}^{n} - \tfrac{1}{2\Delta t}\tilde{\mathbf{u}}^{n-1} - \nabla p^{n+1} \\ &\nabla \cdot \mathbf{u}^{n+1} = 0 \end{aligned} \tag{2.21}$$

Defining: $\tau_0 = \frac{3}{2\Delta t}$, $\tau_1 = -\frac{2}{\Delta t}$, $\tau_2 = \frac{1}{2\Delta t}$ we can rewrite equation (2.21) as:

$$\begin{aligned} &\tau_0\mathbf{u}^{n+1} - \mathcal{D}\mathbf{u}^{n+1} = -\tau_1\tilde{\mathbf{u}}^{n} - \tau_2\tilde{\mathbf{u}}^{n-1} - \nabla p^{n+1} \\ &\nabla \cdot \mathbf{u}^{n+1} = 0 \end{aligned} \tag{2.22}$$

To calculate the velocity $\mathbf{u}^{n+1}$ the pressure-correction method proceeds as follows:

1. *Predict the velocity*
   Using the pressure at the previous time level the approximated velocity ($\mathbf{u}^*$) is calculated as:
   $$\begin{aligned} &\tau_0\mathbf{u}^{*} - \mathcal{D}\mathbf{u}^{*} = -\nabla p^{n} - \tau_1\tilde{\mathbf{u}}^{n} - \tau_2\tilde{\mathbf{u}}^{n-1} \\ &\mathbf{u}^{*} = 0 \text{ on } \Gamma \end{aligned} \tag{2.23}$$

2. *Compute the pressure correction*
   The difference between the actual velocity $\mathbf{u}^{n+1}$ and the approximated one $\mathbf{u}^*$ is given by:
   $$\tau_0(\mathbf{u}^{n+1} - \mathbf{u}^{*}) - (\mathcal{D}\mathbf{u}^{n+1} - \mathcal{D}\mathbf{u}^{*}) = -\nabla(p^{n+1} - p^{n}) \tag{2.24}$$
   Defining $p^* = (p^{n+1} - p^n)$, taking the divergence of both sides of equation (2.24) and neglecting the diffusion terms:
   $$\tau_0 \nabla \cdot (\mathbf{u}^{n+1} - \mathbf{u}^{*}) = -\nabla^2 p^{*} \tag{2.25}$$
   Imposing that the velocity should be divergence free $\nabla \cdot \mathbf{u}^{n+1} = 0$, we obtain the following Poisson equation for the pressure correction:
   $$\nabla^2 p^{*} = \tau_0 \nabla \cdot \mathbf{u}^{*} \tag{2.26}$$

3. *Correct pressure and velocity*
   Correct the velocity according to:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{1}{\tau_0}\nabla p^* \tag{2.27}$$

   Correct the pressure.

$$p^{n+1} = p^* + p^n \tag{2.28}$$

2.2 Spatial discretization

Navier-Stokes equations can be spatially discretized using a Galerkin finite element approach. This is a specific application of the method of weighted residuals, that employs approximating functions for a truncated series expansion of the solution of the partial differential equations. The finite element method utilizes an integral formulation to generate a system of algebraic equations. It uses continuous piecewise smooth functions for approximating the unknown quantity or quantities.

In the finite element discretization, the domain, Ω , is broken up into a set of conformal elements (conformal since all elements connect to neighboring elements through common vertices). The equations are multiplied by a test function and then integrated over the whole domain. This formulation is said to be "weak".

A weak form of the Navier-Stokes equations (2.1) and (2.3), can be derived by introducing weighting functions $\mathbf{w}$ and q in the momentum and continuity equations ($\mathbf{w} \in L^2(\Omega)$ and $q \in L^2(\Omega)$). The pressure can be determined up to a constant which can be fixed by the choice of $q \in Q$.

$$Q = \{q \in L^2(\Omega) | \int_\Omega q \partial\Omega = 0\} \tag{2.29}$$

The weak formulation is

$$\begin{cases} \int_\Omega \rho_F \frac{D\mathbf{u}}{Dt} \cdot \mathbf{w} d\Omega = \int_\Omega \rho_F \mathbf{g} \cdot \mathbf{w} d\Omega + \int_\Omega (\nabla \cdot \boldsymbol{\sigma}) \cdot \mathbf{w} d\Omega \\ \int_\Omega (\nabla \cdot \mathbf{u}) q d\Omega = 0 \end{cases} \tag{2.30}$$

Choosing the weighting functions to be in $\mathbf{H}^1$ ($\mathbf{w} \in \mathbf{H}^1(\Omega)$) equation (2.30) can be integrated by parts and using the tensor identity $(\boldsymbol{\sigma} : \nabla \mathbf{w}) = \nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{w}) - \mathbf{w} \cdot (\nabla \cdot \boldsymbol{\sigma})$ we obtain:

$$\begin{cases} \int_\Omega [\rho_F \frac{D\mathbf{u}}{Dt} \cdot \mathbf{w} + \boldsymbol{\sigma} : (\nabla \mathbf{w})] d\Omega = \int_\Omega \rho_F \mathbf{g} \cdot \mathbf{w} d\Omega + \int_\Gamma (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{w} d\Gamma \\ \int_\Omega (\nabla \cdot \mathbf{u}) q d\Omega = 0 \end{cases} \tag{2.31}$$

where $\mathbf{n}$ is the unit normal. Substituting the stress tensor for the constitutive equation for Newtonian fluids the weak formulations reads:

$$\begin{cases} \int_\Omega (\rho_F \frac{D\mathbf{u}}{Dt} \cdot \mathbf{w} + \mu \mathbf{D}[\mathbf{u}] : \mathbf{D}[\mathbf{w}] - p\nabla \cdot \mathbf{w}) d\Omega = \int_\Omega \rho_F \mathbf{g} \cdot \mathbf{w} d\Omega + \int_\Gamma (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{w} d\Gamma \\ \int_\Omega (\nabla \cdot \mathbf{u}) q d\Omega = 0 \end{cases} \tag{2.32}$$

where $\mathbf{D}$ is rate of strain tensor as defined before. Defining the space of trial functions

$$\mathbf{V} = \{\mathbf{v} | \mathbf{v} \in \mathbf{H}^1(\Omega), \mathbf{v} = \mathbf{g}_0 \text{ on } \Gamma_0\} \tag{2.33}$$

and the space of weighting functions as:

$$\mathbf{W} = \{\mathbf{w} | \mathbf{w} \in \mathbf{H}_0^1(\Omega), \mathbf{w} = 0 \text{ on } \Gamma_0\} \tag{2.34}$$

the weak formulation reads:
Find $\mathbf{v} \in \mathbf{V}$ *and* $p \in Q$ *such that:*

$$\begin{cases} \mathcal{M}(\mathbf{v}, \mathbf{w}) + \mathcal{N}(\mathbf{v}, \mathbf{v}, \mathbf{w}) + \mathcal{S}(\mathbf{v}, \mathbf{w}) + \mathcal{L}(\mathbf{w}, p) = \ell(\mathbf{w}) \\ \mathcal{L}(\mathbf{v}, q) = 0 \; \forall q \in Q \end{cases} \tag{2.35}$$

where:

$$\mathcal{M}(\mathbf{v}, \mathbf{w}) = \int_\Omega \rho_F \frac{\partial \mathbf{v}}{\partial t} \cdot \mathbf{w} d\Omega \tag{2.36}$$

$$\mathcal{N}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \int_\Omega \rho_F [(\mathbf{u} \cdot \nabla)\mathbf{v}] \cdot \mathbf{w} d\Omega \tag{2.37}$$

$$\mathcal{S}(\mathbf{v}, \mathbf{w}) = \int_\Omega \mu \mathbf{D}[\mathbf{v}] : \mathbf{D}[\mathbf{w}] d\Omega \tag{2.38}$$

$$\mathcal{L}(\mathbf{v}, q) = -\int_\Omega (\nabla \cdot \mathbf{v}) q d\Omega \tag{2.39}$$

$$\ell(\mathbf{w}) = \int_\Omega (\mathbf{f} \cdot \mathbf{w}) q d\Omega + \int_\Gamma (\mathbf{g}_1 \cdot \mathbf{w}) q d\Gamma \tag{2.40}$$

A necessary condition for convergence with the integrated and penalty function methods is that the solution space satisfies the Ladyzhenskaya - Babuska - Brezzi (LBB) or divergence - stability condition ([6], [9]):

There exists $\beta > 0$, independent of the mesh spacing, h, such that

$$\sup_{0 \neq \mathbf{v}_h \in \mathbf{V}} \frac{-\int_\Omega (\nabla \cdot \mathbf{v}_h) q d\Omega}{\|\mathbf{v}_h\| \mathbf{v} \| q_h \|_Q} \geq \beta \tag{2.41}$$

for the chosen family of solution spaces, $\mathbf{Z}^h = \{\mathbf{V}, Q\}$. This condition ensures that the discrete divergence free vector field approaches a continuous divergence free field in the limit as $h \to 0$. $\mathbf{Z}$ is the weakly divergence free vector space defined as the set of subspaces satisfying $-\int_\Omega (\nabla \cdot \mathbf{v}) q d\Omega = 0$. In practice the LBB condition implies that the pressure approximation must be taken one or two orders lower than the velocity approximations. If divergence-stability is satisfied and a unique solution exists (conditions are not turbulent), then convergence is assured. As an end result of the LBB condition there is a limit on the acceptable sets of $\mathbf{V}, Q$ since not any combination will satisfy equation (2.41). An admissible element is the second order Taylor-Hood tetrahedron, which is used in our discretization.

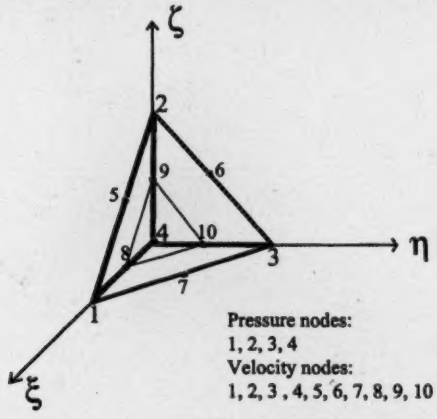

Figure 2.1: Finite element discretization: Taylor-Hood tetrahedron.

Using an appropriate discretization space, the weak form of the Navier-Stokes equations given in equation (2.35) can be rewritten as an algebraic system of equations:

$$\begin{cases} \mathbf{M}\mathbf{v}_h + \mathbf{N}(\mathbf{v}_h)\mathbf{v}_h + \frac{1}{\mathrm{Re}}\mathbf{S}\mathbf{v}_h + \mathbf{L}^T p_h = \mathbf{f} \\ \\ \mathbf{L}\mathbf{v}_h = 0 \end{cases} \tag{2.42}$$

Using the Galerkin formulation, the velocities and pressures are expanded over a finite basis:

$$\mathbf{v}_h = \sum_{i=1}^{N_v} \mathbf{v}_{h,j} \phi_i \tag{2.43}$$

$$p_h = \sum_{i=1}^{N_p} p_{h,i}\psi_i \tag{2.44}$$

Where, ϕ_i and ψ_i are the velocity and pressure finite element basis functions for node i, $\phi_i \in H^1(\Omega)$, $\phi_i|_\Gamma = 0$ and ϕ_i is linearly independent of ψ_i, $\psi \in L^2(\Omega)$. N_v and N_p are the velocity and pressure degrees of freedom. For the element shown in Figure 2.1 the velocity is approximated using 10 nodes and quadratic basis function and the pressure is approximated using 4 nodes (vertices) using linear basis functions.

Introducing the finite dimensional spaces $V_{h,\phi}$ consisting of all linear combinations of $\phi_{i\,i=1}^{N_v}$ and $Q_{h,\psi}$ consisting of all linear combinations of $\psi_{i\,i=1}^{N_p}$ the discrete formulation similar to (2.35) is:

Find $\mathbf{v}_h \in \mathbf{V}_{h,\phi}$ *and* $p_h \in Q_{h,\psi}$ *such that:*

$$\begin{cases} \mathcal{M}(\mathbf{v}_h, \phi_i) + \mathcal{N}(\mathbf{v}_h, \mathbf{v}_h, \phi_i) + \mathcal{D}(\mathbf{v}_h, \phi_i) + \mathcal{L}(\phi_i, p_h) = \ell(\phi_i),\ i = 1, ..., N_v \\ \\ \mathcal{L}(\mathbf{v}_h, \psi_j) = 0,\ j = 1, ..., N_p \end{cases} \tag{2.45}$$

Substituting for $\mathbf{v}_h$, p_h, ϕ_i and ψ_i in equations (2.36)-(2.40) and dropping the subscript h we can rewrite the Navier-Stokes equations in semi-discrete form and obtain the following system of differential-algebraic equations:

$$\begin{cases} \mathbf{M}\dot{\mathbf{v}} + \mathbf{N}(\mathbf{v})\mathbf{v} + \frac{1}{\mathrm{Re}}\mathbf{S}\mathbf{v} + \mathbf{L}^T p = \mathbf{f} \\ \\ \mathbf{L}\mathbf{v} = 0 \end{cases} \tag{2.46}$$

The matrices $\mathbf{M}$ (mass matrix), $\mathbf{N}$ (advection matrix), $\mathbf{S}$ (stiffness or Laplacian matrix) and $\mathbf{L}$ (gradient matrix) are given by:

$$M_{i,j} = \int_\Omega \phi_i \cdot \phi_j d\Omega \tag{2.47}$$

$$S_{i,j} = \int_\Omega \nabla\phi_i \cdot \nabla\phi_j d\Omega \tag{2.48}$$

$$N_{k,l} = \sum_{m=1}^{N_v} \mathbf{v}_k \int_\Omega \phi_k \phi_l \nabla\phi_l d\Omega \tag{2.49}$$

$$L^i_{k,l} = -\int_\Omega \phi_k \frac{d\psi_l}{dx_i} d\Omega \tag{2.50}$$

where Re is the Reynolds number and $\mathbf{f}$ is a vector that includes the boundary conditions. The mass and stiffness matrix are symmetric and positive definite. The transpose of the gradient matrix is the negative of the divergence matrix.
The initial conditions are given by:

$$\mathbf{v}(0) = \mathbf{v}_0 \text{ with the constraint that } [\mathbf{L}]\mathbf{v}_0 = 0 \tag{2.51}$$

2.3 Navier-Stokes solver validation

A finite element 3-D Navier-Stokes solver was implemented using the discretization scheme described in the previous sections. The spatial discretization was done using $P_2 - P_1$ finite elements. An operator splitting technique was used for the time discretization. Within this method, the convection part of the Navier-Stokes equations was solved using the method of Characteristics. The generalized Stokes problem was solved using a pressure-correction scheme. The code was validated using some benchmark test cases. Two of them are presented here: backward facing step flow and lid-driven cavity.

2.3.1 Case 1: Backward facing step

The flow in a channel over a backward facing step is used as a benchmark to evaluate the accuracy of numerical schemes. Figure 2.2 shows a sketch of this case problem. The channel is defined to have a height of H, a width of $0.4H$ and a length of $15H$. The upstream inlet area is half of the outlet ($0.5H \times 0.4H$). One of the main features of the backward facing step flow is a recirculation region just downstream of the step. At the enlargement, the flow velocity is suddenly reduced and as a consequence the pressure is increased. The governing equations for a two dimensional stationary flow are:

Continuity equation

$$\frac{\partial(\rho_F u_x)}{\partial x} + \frac{\partial(\rho_F u_y)}{\partial y} = 0 \tag{2.52}$$

Momentum equation

$$\frac{\partial(\rho_F u_x^2)}{\partial x} + \frac{\partial(\rho_F u_x u_y)}{\partial y} = -\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2}\right) \tag{2.53}$$

$$\frac{\partial(\rho_F u_x u_y)}{\partial x} + \frac{\partial(\rho_F u_y^2)}{\partial y} = -\frac{\partial p}{\partial y} + \mu\left(\frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2}\right) \tag{2.54}$$

Just downstream of the sudden enlargement a recirculation area is formed. The length of this recirculation region is a function of the geometry (enlargement ratio) and the Reynolds number (fluid momentum and regime: laminar or turbulent). In the current simulation we want to compare our predictions with numerical and experimental results available in the literature.

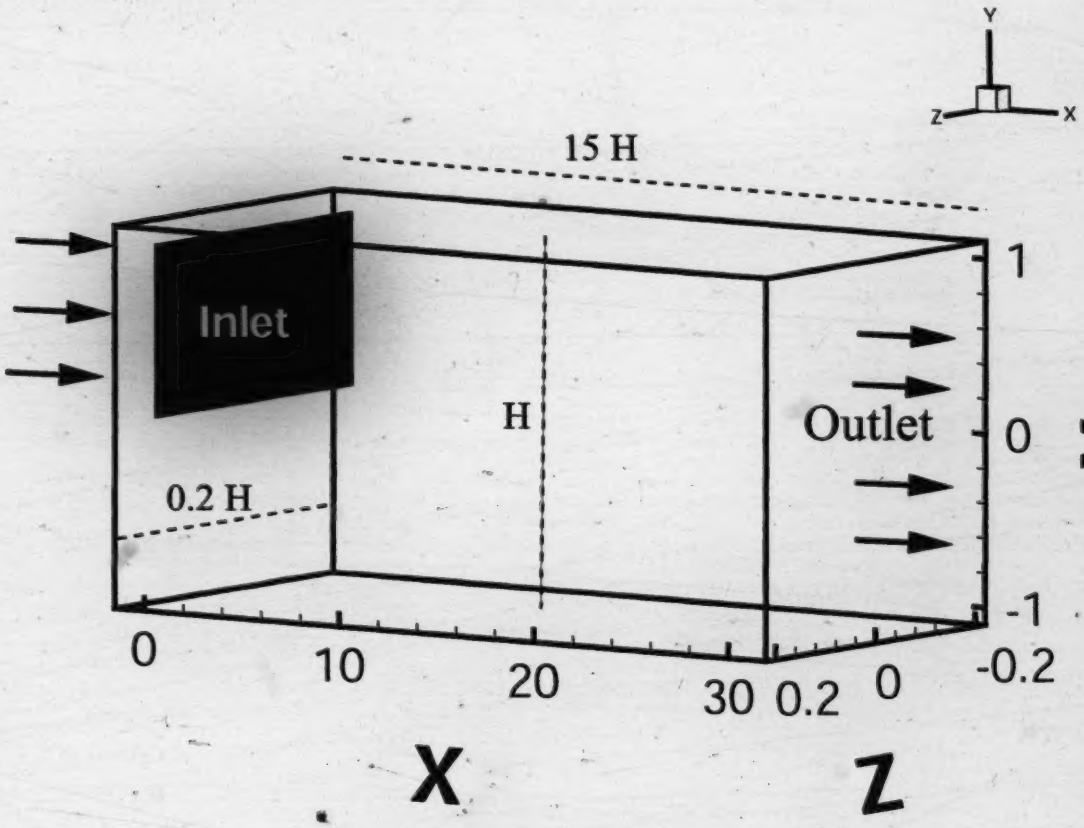

Figure 2.2: Test case: Backward facing step.

The mesh for the simulation was incrementally refined until there was no variation in the solution with further mesh refinement. Here we show the solution obtained using a mesh with 22525 nodes and 11800 elements. The channel's geometry is shown in Figure 2.2. Its dimensions are: $2 \times 0.8 \times 30$ (height $\times$ width $\times$ length). Boundary

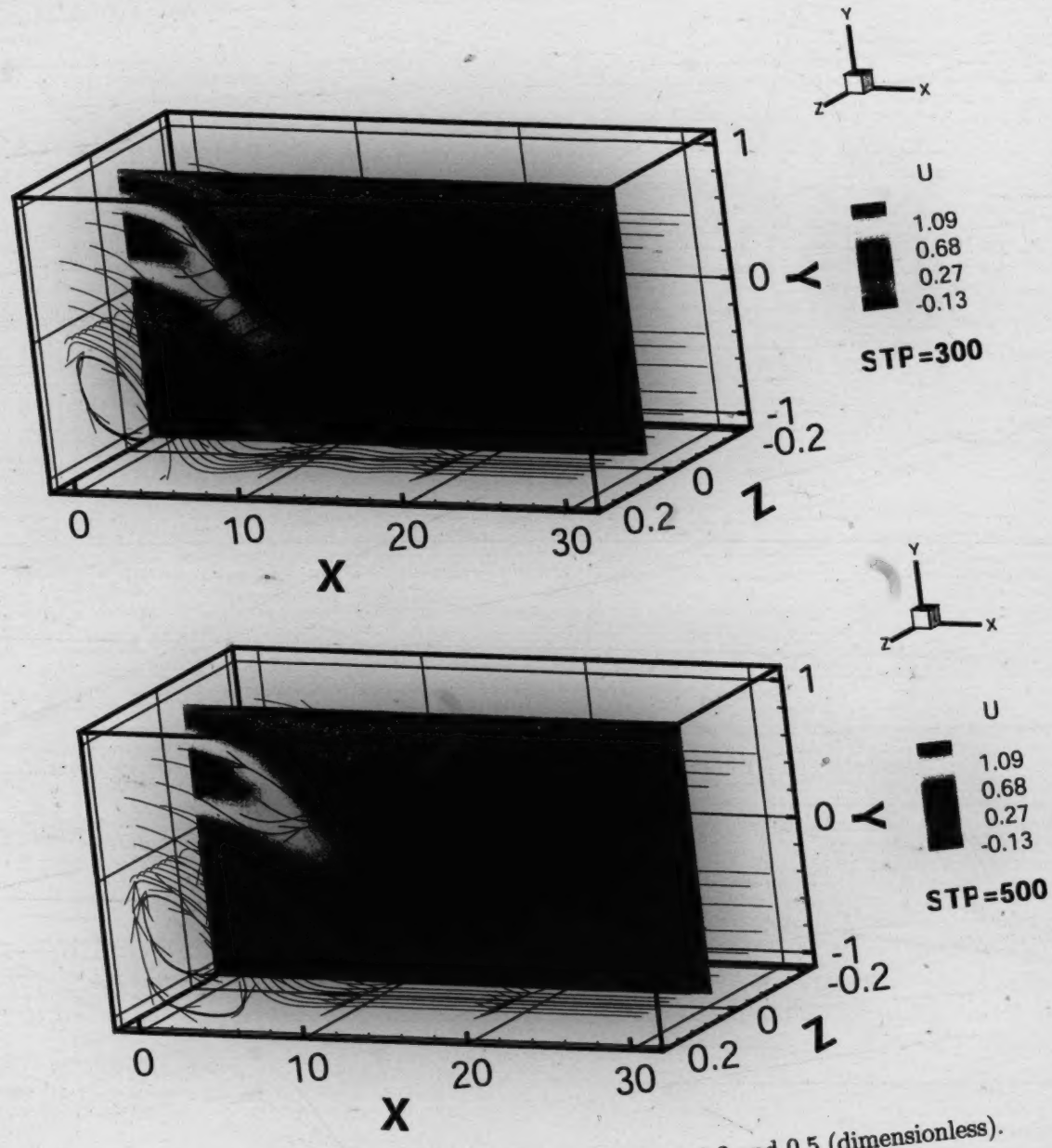

Figure 2.3: Backward facing step. Re=100, t=0.3 and 0.5 (dimensionless).

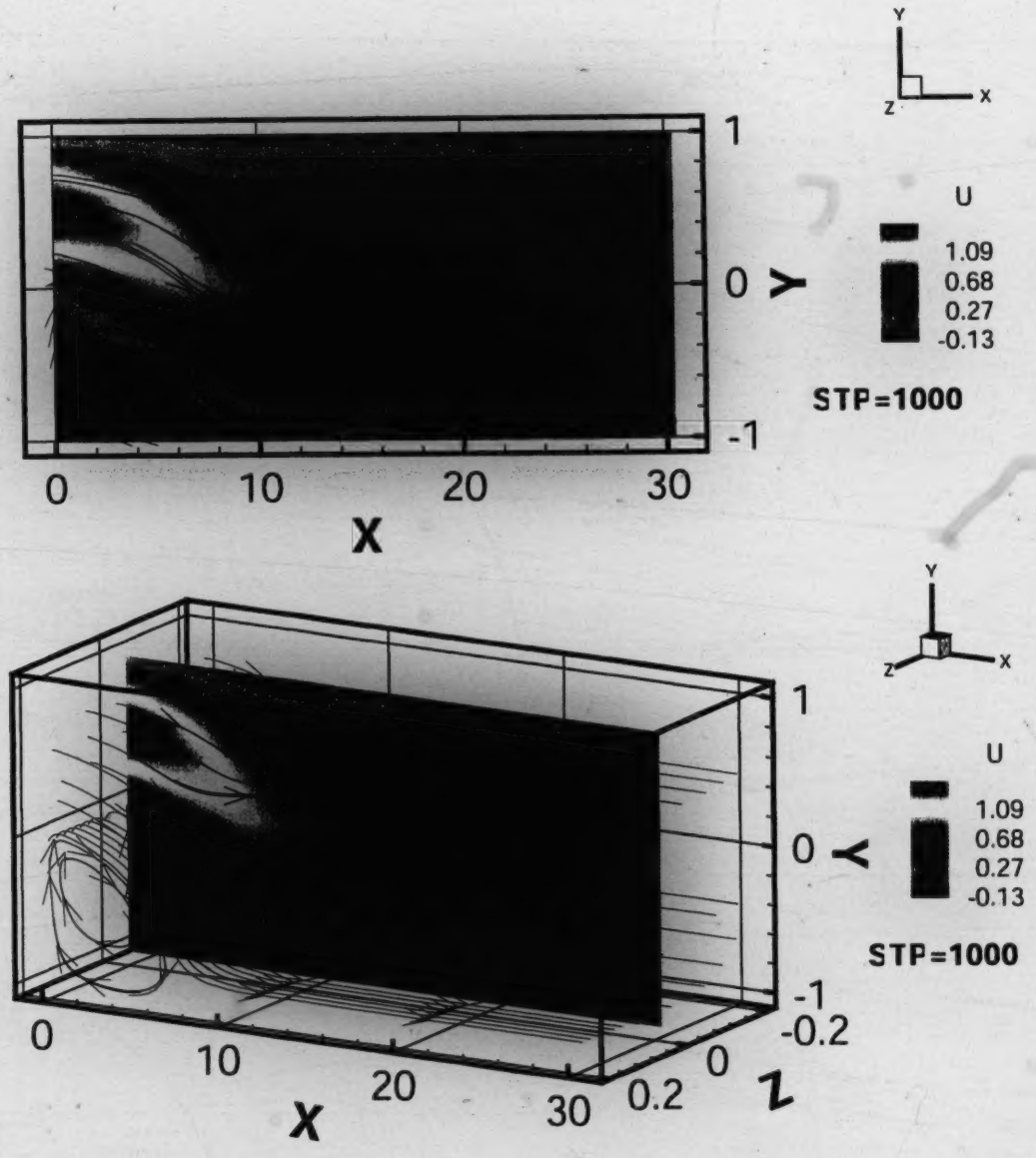

Figure 2.4: Backward facing step. Re=100, t=1.0 (dimensionless).

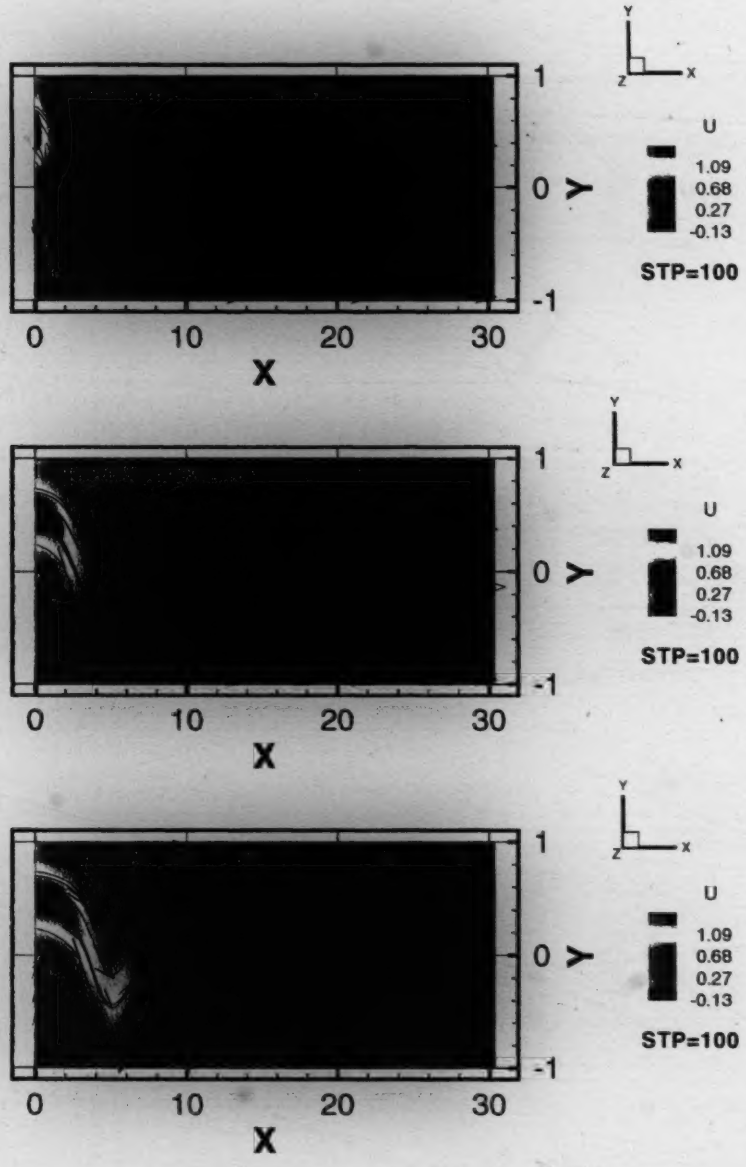

Figure 2.5: Backward facing step. Re=1, 50 and 100, t=0.1 (dimensionless).

conditions were prescribed at all the walls: no-slip condition for the top and bottom walls (2415 nodes) and symmetry condition for the side walls (9954 nodes). The inlet flows through the upper half portion of the front wall (55 nodes) with a prescribed parabolic horizontal component given by:

$$\mathbf{u}_x(y) = 6.0y(1.0 - y) \text{ for } 0.0 \leq y \leq 1.0 \tag{2.55}$$

This profile produces a maximum inflow velocity of $u_{\max} = 1.5$ and an average velocity of $u_{\text{average}} = 1$. The outflow boundary condition (105 nodes) is of a constant stress normal to the boundary. The normal stress condition can be written as:

$$-p + 2\mu\frac{\partial \mathbf{u}}{\partial x} = \text{ constant} \tag{2.56}$$

In this case the constant was set to zero. This is equivalent to setting the pressure to zero since the second term of equation (2.56) is negligible at the end of the channel. Initially the fluid is at rest and a parabolic profile is prescribed at the inlet. The Reynolds number for this simulation was defined as:

$$\text{Re} = \frac{u_{\text{average}} D_c}{\nu} \tag{2.57}$$

where u_{average} is the mean inlet velocity, D_c is the hydraulic diameter of the inlet (height of the backward step) and ν is the kinematic viscosity of the fluid. Figures 2.3-2.5 show the streamtraces obtained for different Reynolds numbers. Our numerical results were compared to those of Kim et al. [48], Armaly et al. [4] and Ethier et al.[54]. The results matched the ones of Ethier et al. [54] very well (less than 1.5% difference). As pointed out by Armaly et al. [4] for smaller Re (less than 400) the flow shows only a primary region of separated flow attached to the backward facing step. The length of this separated flow region increase non-linearly with the Reynolds number. For the case of Re=100, the recirculation length was compared to that of previous studies. In our case the dimensionless length of the recirculation zone was of 3.25 compared to 3.29 obtained by Ethier et al. in their numerical simulations, 3.2 obtained by Kim et al. in their 2-D simulation and 3.2 measured by Armaly et al. in their experimental work.

2.3.2 Case 2: Lid-driven cavity flow

The one-sided lid-driven cavity flow is also a benchmark test in numerical simulations. This flow problem contains discontinuous boundary conditions in the corner regions

where the moving lid meets the stationary walls. Figure 2.6 shows the system being considered. In our case the domain is a cube of unit height, length and width. The conditions for the velocity are no-slip at the walls and constant velocity for the lid.

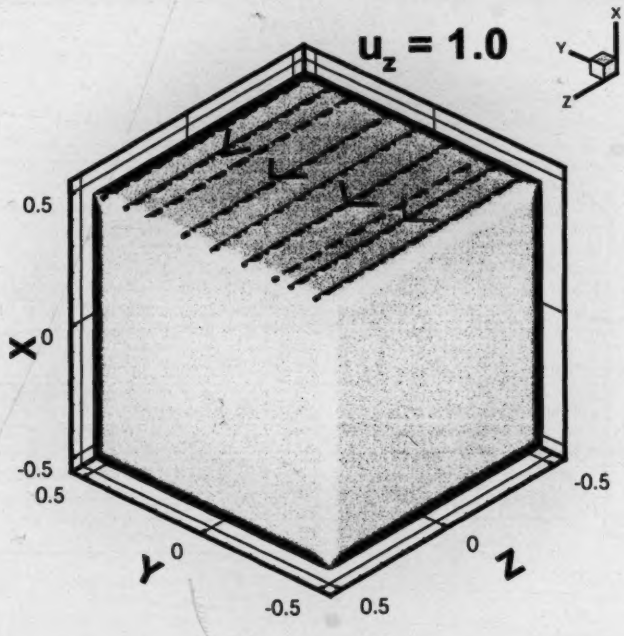

Figure 2.6: Lid-driven cavity test case.

The mesh used for this simulation had 60921 nodes and 40000 elements. The time step was set to 0.0001 and the Re=100. The total duration of the run was 2000 time steps. No-slip boundary condition is set for five of the walls. A constant velocity is prescribed for the lid (top wall) with $u_z = 1$. Figures 2.7 and 2.8 show the simulation domain and the streamtraces as the flow evolves with time. The results of these simulation are compared to that of Ethier et al. [54] and Iwatsu el al. [43]. Figure 2.9 shows the dimensionless velocity in the z direction along the centerline of the cubic cavity ($y = 0, z = 0, -0.5 \leq z \leq 0.5$). The results from our Navier-Stokes equations solver (NSE solver) are compared to the results presented by these authors. The difference of our numerical experiments and [54] was less than 1% and about 2% with respect to [43].

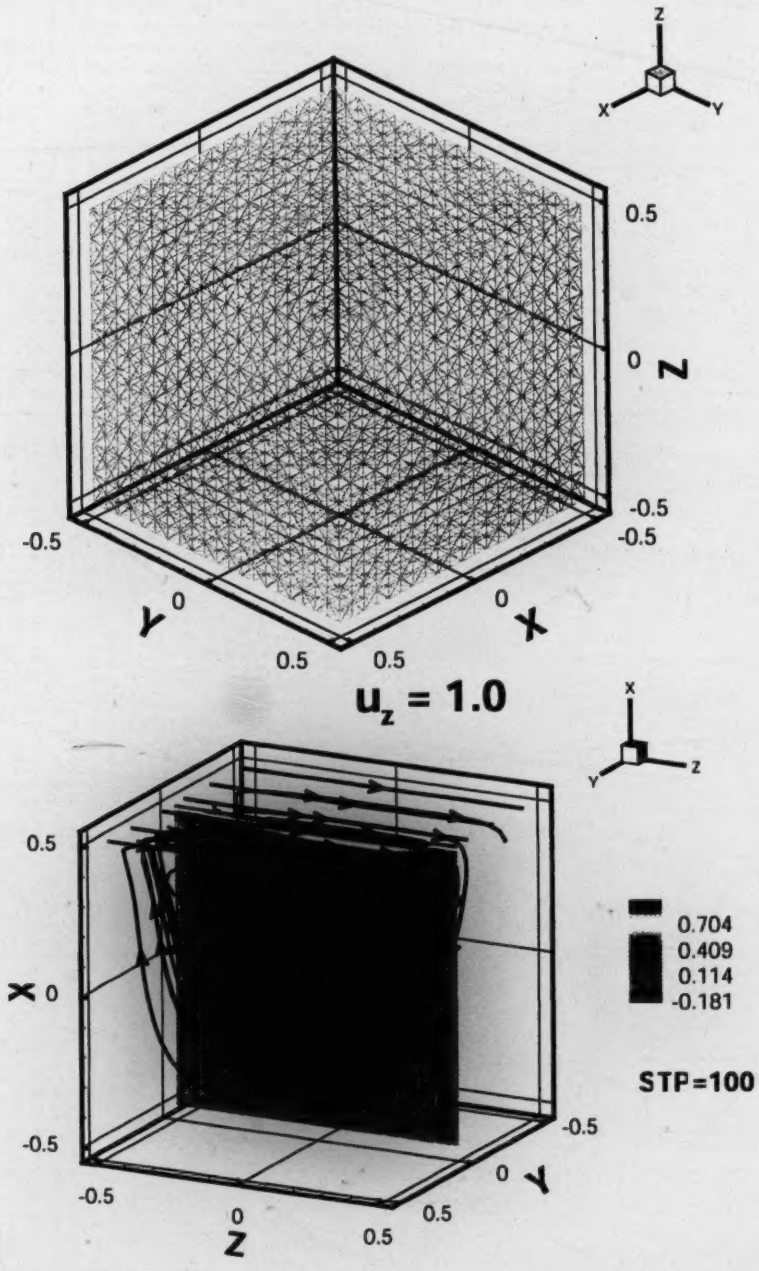

Figure 2.7: Lid-driven cavity test case. a) Domain b) Re=100, t=0.01 (dimensionless).

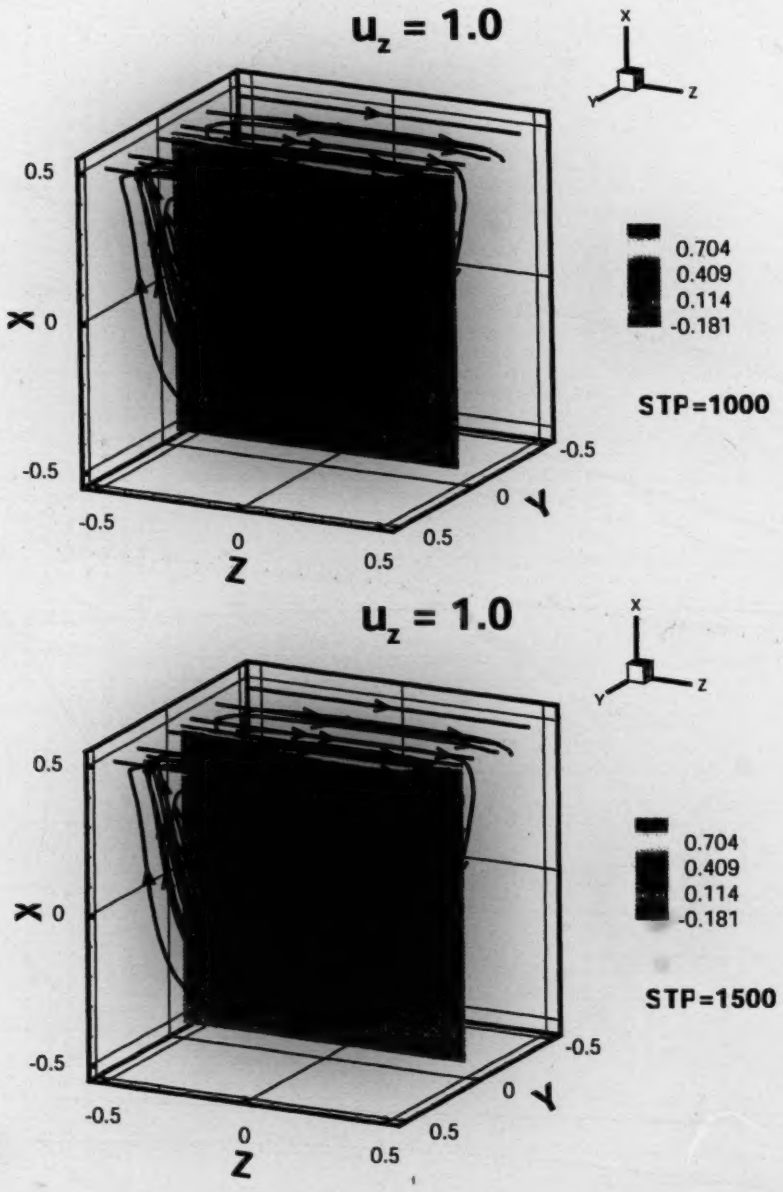

Figure 2.8: Lid-driven cavity test case. Re=100, t=1.5 (dimensionless).

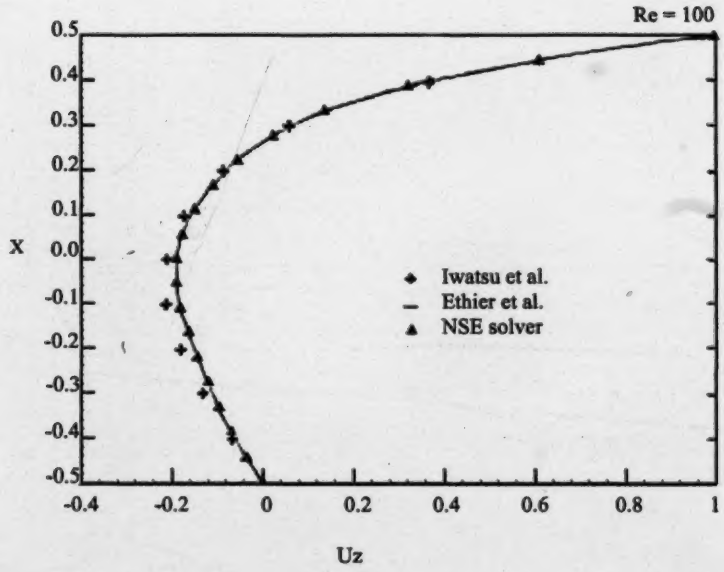

Figure 2.9: Lid-driven cavity. Comparison of velocity results (dimensionless) along the centerline of the cavity for Re=100.

2.4 Conclusions

In this chapter the numerical methods used to solve the Navier-Stokes equations for incompressible unsteady flow were presented. Galerkin finite elements were used for spatial discretization. A operator splitting technique was used for time discretization. This allowed the Stokes and continuity equations to be solved separately from the convective terms, using an adequate time integration scheme for each operator. The convection problem was solved using the method of characteristics. The generalized Stokes problem was discretized using a projection method, in particular a pressure correction algorithm second order accurate in time. This pressure-correction method allowed us to decouple the pressure and the velocity. The algorithm was implemented into a code and this was validated using some available data from the literature. Two benchmark cases were presented: lid-driven cavity and backward facing step. Results from our simulations were in good agreement with the experimental data.

Chapter 3

A Lagrange multiplier/fictitious domain formulation

3.1 Governing equations

For a system of an incompressible Newtonian fluid and rigid solid particles, let's consider a domain denoted by Ω with boundary Γ with n solid particles denoted as P_i with boundary $\partial\Omega$ (see Figure 3.1). The region occupied by the particles is $P(t) = \cup_{i=1}^{n} P_i(t)$.

The equations that govern the **fluid motion** can then be written as:

Momentum equation

$$\rho_F \frac{D\mathbf{u}}{Dt} = \rho_F \mathbf{g} + \nabla \cdot \boldsymbol{\sigma} \text{ in } \Omega \backslash \overline{P(t)} \tag{3.1}$$

Mass Conservation Equation

The continuity equation for an incompressible fluid is:

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \backslash \overline{P(t)} \tag{3.2}$$

where $\mathbf{u}$ is the velocity of the fluid, ρ_F is the density of the fluid, $\frac{D\mathbf{u}}{Dt}$ is the material derivative and $\boldsymbol{\sigma}$ is the stress tensor. The boundary and initial conditions are:

$$\mathbf{u} = \mathbf{u}_\Gamma(t) \text{ on } \Gamma \tag{3.3}$$

$$\mathbf{u} = \mathbf{U}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i \text{ on } \partial\Omega_i(t) \qquad i = 1...n \tag{3.4}$$

$$\mathbf{u}|_{t=0} = \mathbf{u}_0 \text{ in } \Omega \backslash \overline{P(0)} \tag{3.5}$$

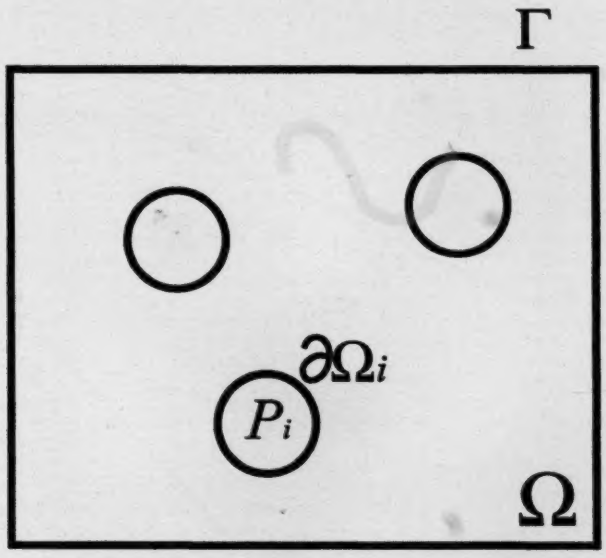

Figure 3.1: Solid particles in a fluid domain.

where $\mathbf{U}_i$ is the ith particle's velocity, $\boldsymbol{\omega}_i$ is its angular velocity, and $\mathbf{r}_i$ is its radius.

The corresponding **rigid body** equations are:

Newton's second law:

$$M_i \frac{d\mathbf{U}_i}{dt} = M_i \mathbf{g} + \mathbf{F}_i \tag{3.6}$$

Euler's rotation equation:

$$\mathbf{I}_i \frac{d\boldsymbol{\omega}_i}{dt} + \boldsymbol{\omega}_i \times \mathbf{I}_i \boldsymbol{\omega}_i = \mathbf{T}_i \tag{3.7}$$

Particle's position and orientation:

$$\frac{d\mathbf{X}_i}{dt} = \mathbf{U}_i \tag{3.8}$$

$$\frac{d\theta_i}{dt} = \boldsymbol{\omega}_i \tag{3.9}$$

where $\mathbf{I}_i, \mathbf{U}_i, \boldsymbol{\omega}_i$ are the moment of inertia, translational velocity and angular velocity of the ith particle, $\mathbf{F}_i$ and $\mathbf{T}_i$ are the hydrodynamic force and torque about the center of mass on the ith particle and $\mathbf{g}$ is the acceleration of gravity.

The hydrodynamic force and torque can also be expressed as:

$$\mathbf{F}_i = -\int_{\partial\Omega_i} \boldsymbol{\sigma} \cdot \mathbf{n} dS \tag{3.10}$$

$$\mathbf{T}_i = -\int_{\partial\Omega_i} \mathbf{r}_i \times (\boldsymbol{\sigma}\cdot\mathbf{n})dS \tag{3.11}$$

with $\mathbf{n}$ the unit normal vector pointing out of the particle.
The boundary and initial conditions can be stated as:

$$\mathbf{U}_i|_{t=0} = \mathbf{U}_{i,0} \tag{3.12}$$
$$\boldsymbol{\omega}_i|_{t=0} = \boldsymbol{\omega}_{i,0} \tag{3.13}$$
$$\mathbf{X}_i|_{t=0} = \mathbf{X}_{i,0} \tag{3.14}$$
$$\theta_i|_{t=0} = \theta_{i,0} \tag{3.15}$$

3.2 Weak formulation in the combined fluid-particle space

Following Glowinski ([34], [32], [31]) a weak form of the governing equations can be derived. The hydrodynamic forces and torques that appear on the rigid body equation can be eliminated by combining the fluid and particle equations of motion and defining a combined velocity space. For simplicity, the following formulation is presented for one solid particle. Its extension to take into account many solid particles is straightforward.

We introduce the functional space of test functions and define the test functions for the fluid and the particle, and impose that on the particle's boundary, the fluid and particle velocity should have the same value.

$$\begin{aligned}\mathbf{V}(t) = \{&(\widehat{\mathbf{v}},\widehat{\mathbf{V}},\widehat{\xi})|\widehat{\mathbf{v}}\in H^1(\Omega\backslash\overline{P(t)})^3, \widehat{\mathbf{V}}\in\mathbf{R}^3, \widehat{\xi}\in\mathbf{R}^3,\\ &\widehat{\mathbf{v}} = \widehat{\mathbf{V}} + \widehat{\xi}\times\mathbf{r} \text{ on } \partial\Omega(t), \text{ and } \widehat{\mathbf{v}} = \mathbf{u}_\Gamma(t) \text{ on } \Gamma\}\end{aligned} \tag{3.16}$$

Multiplying the fluid momentum equation by the test functions and integrating over the fluid domain we obtain:

$$\int_{\Omega\backslash\overline{P(t)}} \rho_F \frac{D\mathbf{u}}{Dt}\cdot\widehat{\mathbf{v}}d\Omega + \int_{\Omega\backslash\overline{P(t)}} \rho_F\mathbf{g}\cdot\widehat{\mathbf{v}}d\Omega = \int_{\Omega\backslash\overline{P(t)}} (\nabla\cdot\boldsymbol{\sigma})\cdot\widehat{\mathbf{v}}d\Omega \tag{3.17}$$

Using the tensor identity [3]:

$$\tau : \mathbf{D}[\mathbf{v}] = \nabla\cdot(\tau\cdot\mathbf{v}) - (\nabla\cdot\tau)\cdot\mathbf{v} \tag{3.18}$$

and integrating, the right hand side of equation (3.17) by parts we obtain:

$$\int_{\Omega\backslash\overline{P(t)}} (\nabla\cdot\boldsymbol{\sigma})\cdot\widehat{\mathbf{v}}d\Omega = \int_{\Omega\backslash\overline{P(t)}} \nabla\cdot(\boldsymbol{\sigma}\cdot\widehat{\mathbf{v}}) - \int_{\Omega\backslash\overline{P(t)}} \boldsymbol{\sigma} : \mathbf{D}[\widehat{\mathbf{v}}]d\Omega \tag{3.19}$$

For a volume V, with a closed surface S and **a** any vector field defined on V and on S, Green's theorem states that if S is piece-wise smooth with outward normal **n** and **a** continuously differentiable then:

$$\iiint_V \nabla \cdot \mathbf{a} = \iint_S \mathbf{a} \cdot \mathbf{n} dS \tag{3.20}$$

Applying Green's theorem (divergence Theorem) to equation (3.19) we get:

$$\int_{\Omega \setminus \overline{P(t)}} (\nabla \cdot \boldsymbol{\sigma}) \cdot \widehat{\mathbf{v}} d\Omega = \int_{\partial\Omega} (\boldsymbol{\sigma} \cdot \widehat{\mathbf{v}}) \cdot \mathbf{n} dS - \int_{\Omega \setminus \overline{P(t)}} \boldsymbol{\sigma} : \mathbf{D}[\widehat{\mathbf{v}}] d\Omega \tag{3.21}$$

Substituting equation (3.21) in equation (3.17):

$$\int_{\Omega \setminus \overline{P(t)}} \rho_F \frac{D\mathbf{u}}{Dt} \cdot \widehat{\mathbf{v}} d\Omega + \int_{\Omega \setminus \overline{P(t)}} \rho_F \mathbf{g} \cdot \widehat{\mathbf{v}} d\Omega = \int_{\partial\Omega} (\boldsymbol{\sigma} \cdot \widehat{\mathbf{v}}) \cdot \mathbf{n} dS - \int_{\Omega \setminus \overline{P(t)}} \boldsymbol{\sigma} : \mathbf{D}[\widehat{\mathbf{v}}] d\Omega \tag{3.22}$$

Multiplying the particle's motion equation by the test functions:

$$M_s(\frac{d\mathbf{U}}{dt} - \mathbf{g}) \cdot \widehat{\mathbf{V}} = -\int_{\partial\Omega} \boldsymbol{\sigma} \cdot \mathbf{n} \cdot \widehat{\mathbf{V}} dS \tag{3.23}$$

$$\mathbf{I}_s \frac{d\boldsymbol{\omega}}{dt} \cdot \widehat{\boldsymbol{\xi}} + (\boldsymbol{\omega} \times \mathbf{I}_s \boldsymbol{\omega}) \cdot \widehat{\boldsymbol{\xi}} = -\int_{\partial\Omega} \mathbf{r} \times (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \widehat{\boldsymbol{\xi}} dS \tag{3.24}$$

But on the sphere's boundary $\partial\Omega$, $\widehat{\mathbf{v}} = \widehat{\mathbf{V}} + \widehat{\boldsymbol{\xi}} \times \mathbf{r}$. Substituting $\widehat{\mathbf{v}}$ in the right side of equation (3.22) we obtain:

$$\int_{\Omega \setminus \overline{P(t)}} (\nabla \cdot \boldsymbol{\sigma}) \cdot \widehat{\mathbf{v}} d\Omega = \int_{\partial\Omega} (\boldsymbol{\sigma} \cdot \widehat{\mathbf{V}}) \cdot \mathbf{n} dS + \int_{\partial\Omega} \boldsymbol{\sigma} \cdot (\widehat{\boldsymbol{\xi}} \times \mathbf{r}) \cdot \mathbf{n} dS - \int_{\Omega \setminus \overline{P(t)}} \boldsymbol{\sigma} : \mathbf{D}[\widehat{\mathbf{v}}] d\Omega \tag{3.25}$$

Using the identity $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a})$ we can rewrite (3.25) as:

$$\int_{\Omega \setminus \overline{P(t)}} (\nabla \cdot \boldsymbol{\sigma}) \cdot \widehat{\mathbf{v}} d\Omega = \int_{\partial\Omega} (\boldsymbol{\sigma} \cdot \widehat{\mathbf{V}}) \cdot \mathbf{n} dS + \int_{\partial\Omega} \widehat{\boldsymbol{\xi}} \cdot (\mathbf{r} \times \boldsymbol{\sigma} \cdot \mathbf{n}) dS - \int_{\Omega \setminus \overline{P(t)}} \boldsymbol{\sigma} : \mathbf{D}[\widehat{\mathbf{v}}] d\Omega \tag{3.26}$$

Adding equations (3.22),(3.23) and (3.24) and using equation (3.26) we obtain the combined equation of motion:

$$\int_{\Omega \setminus \overline{P(t)}} \rho_F \frac{D\mathbf{u}}{Dt} \cdot \widehat{\mathbf{v}} d\Omega + \int_{\Omega \setminus \overline{P(t)}} \rho_F \mathbf{g} \cdot \widehat{\mathbf{v}} d\Omega + M_s(\frac{d\mathbf{U}}{dt} - \mathbf{g}) \cdot \widehat{\mathbf{V}} +$$
$$\mathbf{I}_s \frac{d\boldsymbol{\omega}}{dt} \cdot \widehat{\boldsymbol{\xi}} + (\boldsymbol{\omega} \times \mathbf{I}_s \boldsymbol{\omega}) \cdot \widehat{\boldsymbol{\xi}} = -\int_{\Omega \setminus \overline{P(t)}} \boldsymbol{\sigma} : \mathbf{D}[\widehat{\mathbf{v}}] d\Omega$$

Requiring that the pressure p lie in the $L^2_0(\Omega\backslash\overline{P(t)})$ space, the weak form of the incompressibility constraint is:

$$L^2_0(\Omega\backslash\overline{P(t)}) = \{q \in L^2(\Omega\backslash\overline{P(t)}) | \int_{\Omega\backslash\overline{P(t)}} q d\Omega = 0\} \tag{3.27}$$

$$\int_{\Omega\backslash\overline{P(t)}} q\nabla \cdot \mathbf{u} d\Omega = 0 \text{ for all } q \in L^2(\Omega\backslash\overline{P(t)}) \tag{3.28}$$

The weak formulation is completed with the equations corresponding to the particle's position, equations (3.8) and (3.9), together with the boundary and initial conditions, equations (3.3), (3.4),(3.5) and (3.15).

3.3 Fictitious domain formulation

To obtain a fictitious domain formulation the problem has to be restated with equations that are valid not only in $\Omega\backslash\overline{P(t)}$ but in the entire domain. This will allow us to use a fixed mesh for the flow calculations simplifying greatly the computation cost. In such scenario the rigid bodies can be thought as if they were filled with the surrounding fluid that behaves as a rigid body. Lagrange multipliers will enforce the fluid-solid interaction (hydrodynamic forces and torques) forcing the solution to satisfy the no-slip condition on the particle's boundary.

In order to extend the weak formulation to the entire domain the rigid body constraint is enforced throughout the entire solid particle [32]:

$$\begin{cases} \mathbf{u} = \mathbf{U} + \boldsymbol{\omega} \times \mathbf{r} \text{ in } P(t) \\ \widehat{\mathbf{v}} = \widehat{\mathbf{V}} + \widehat{\boldsymbol{\xi}} \times \mathbf{r} \text{ in } P(t) \end{cases} \tag{3.29}$$

This implies:

$$\frac{D\mathbf{u}}{dt} = \frac{d\mathbf{U}}{dt} + \frac{d\boldsymbol{\omega}}{dt} \times \mathbf{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \text{ in } P(t) \tag{3.30}$$

and

$$\int_{P(t)} \rho_s \frac{D\mathbf{u}}{dt} \cdot \widehat{\mathbf{v}} d\Omega = M_s \mathbf{g} \cdot \widehat{\mathbf{V}} \tag{3.31}$$

Taking the inner product of equation (3.30) with $\rho_s \widehat{\mathbf{v}}$, integrating over the particle $P(t)$, using the boundary condition on the particle and subtracting equation (3.31) from this, we obtain:

$$\int_{P(t)} \rho_s (\frac{D\mathbf{u}}{dt} - \mathbf{g}) \cdot \widehat{\mathbf{v}} d\Omega = M_s (\frac{d\mathbf{U}}{dt} - \mathbf{g}) \cdot \widehat{\mathbf{V}} + (\mathbf{I}\frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}) \cdot \widehat{\boldsymbol{\xi}} \tag{3.32}$$

The combined weak equation of motion can be now extended to the entire domain Ω using equation (3.29). The extended combined variational space is then:

$$\begin{aligned}\widehat{\mathbb{V}}(t) = \{(\widehat{\mathbf{v}}, \widehat{\mathbf{V}}, \widehat{\xi}) | \widehat{\mathbf{v}} \in H^1(\Omega)^3, \widehat{\mathbf{V}} \in \mathbf{R}^3, \widehat{\xi} \in \mathbf{R}^3, \\ \widehat{\mathbf{v}} = \widehat{\mathbf{V}} + \widehat{\xi} \times \mathbf{r} \text{ in } P(t), \text{ and } \widehat{\mathbf{v}} = 0 \text{ on } \Gamma\}\end{aligned} \tag{3.33}$$

Multiplying (3.32) by ρ_F/ρ_s, adding it to (3.27) and noting that $\mathbf{D}[\widehat{\mathbf{v}}] = 0$ in $P(t)$ we obtain the combined weak equation of motion for the entire domain:

$$\begin{aligned}&\int_\Omega \rho_F(\frac{D\mathbf{u}}{Dt} - \mathbf{g}) \cdot \widehat{\mathbf{v}} d\Omega + (1 - \frac{\rho_F}{\rho_s})(M_s(\frac{d\mathbf{U}}{dt} - \mathbf{g}) \cdot \widehat{\mathbf{V}} + (\mathbf{I}_s \frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\omega} \times \mathbf{I}_s\boldsymbol{\omega}) \cdot \widehat{\xi}) \\ &= -\int_\Omega \boldsymbol{\sigma} : \mathbf{D}[\widehat{\mathbf{v}}] d\Omega, \quad \forall (\widehat{\mathbf{v}}, \widehat{\mathbf{V}}, \widehat{\xi}) \in \widehat{\mathbb{V}}\end{aligned} \tag{3.34}$$

The rigid body constraint (enforced through the definition of the combined velocity space) can be relaxed by enforcing it in a weak sense as a side constraint using Lagrange multipliers (see [32]). The problem can be restated as:

For $t > 0$ find $\mathbf{u} \in \mathbb{W}$ with $\mathbb{W} = \{\widehat{\mathbf{v}} \in H^1(\Omega)^3 | \widehat{\mathbf{v}}\}$, $p \in L^2_0(\Omega)$ with $L^2_0(\Omega) = \{q \in L^2(\Omega) | \int_\Omega q d\Omega = 0\}$, $\mathbf{U} \in \mathbf{R}^3$, $\boldsymbol{\omega} \in \mathbf{R}^3$, $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$ with $\boldsymbol{\Lambda} = H^1(P(t))^2$ *satisfying:*

$$\begin{cases} \int_\Omega \rho_F(\frac{D\mathbf{u}}{Dt} - \mathbf{g}) \cdot \widehat{\mathbf{v}} d\Omega - \int_\Omega p \nabla \cdot \widehat{\mathbf{v}} d\Omega + \int_\Omega 2\mu \mathbf{D}[\mathbf{u}] : \mathbf{D}[\widehat{\mathbf{v}}] d\Omega \\ + (1 - \frac{\rho_F}{\rho_s})(M_s(\frac{d\mathbf{U}}{dt} - \mathbf{g}) \cdot \widehat{\mathbf{V}} + (\mathbf{I}_s \frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\omega} \times \mathbf{I}_s\boldsymbol{\omega}) \cdot \widehat{\xi}) \\ = \langle \boldsymbol{\lambda}, \widehat{\mathbf{v}} - (\widehat{\mathbf{V}} + \widehat{\xi} \times \mathbf{r}) \rangle_{P(t)} \\ \forall \widehat{\mathbf{v}} \in \mathbb{W}, \widehat{\mathbf{V}} \in \mathbf{R}^3 \text{ and } \widehat{\xi} \in \mathbf{R}^3 \\ \\ \int_\Omega q \nabla \cdot \mathbf{u} d\Omega = 0 \quad \forall q \in L^2(\Omega) \\ \\ \langle \alpha, \mathbf{u} - (\mathbf{U} + \boldsymbol{\omega} \times \mathbf{r}) \rangle_{P(t)} = 0 \; \forall \alpha \in \boldsymbol{\Lambda}(t) \end{cases} \tag{3.35}$$

With initial conditions:

$$\begin{cases} \nabla \cdot \mathbf{u}_0 = 0 \text{ in } \Omega \backslash \overline{P(0)} \\ \\ \mathbf{u}_0 = \mathbf{U}_0 + \boldsymbol{\omega}_0 \times \mathbf{r} \text{ in } P(0) \end{cases} \tag{3.36}$$

Here $\langle \cdot, \rangle_{P(t)}$ is an appropriate inner product. For the current algorithm this was chosen to be the standard inner product on $H^1(P(t))^3$. In Chapter 4 the choice of the inner product is discussed in more detail.

In the above formulation, the Lagrange multipliers can be seen as additional body force per unit volume that accounts for the fluid-particle interaction and enforces the rigidity inside the particle.

3.4 Numerical Scheme

3.4.1 Finite Element Discretization

Equations (3.35) and (3.36) are spatially discretized using a Galerkin finite element approach as described in Chapter 2. Given a discretized velocity and pressure domain, $\mathbb{D}$, and a space step in that discretized domain, d, we can define the finite dimensional space approximating $\mathbb{W}$, and $L^2(\Omega)$:

$$\mathbb{W}_d = \{\mathbf{v}_d \mid \mathbf{v}_d \in (C^0(\Omega))^3,\ \mathbf{v}_d \in P_2,\ \forall D \in \mathbb{D}_d\} \tag{3.37}$$

$$\mathbb{W}_{0d} = \{\mathbf{v}_d \mid \mathbf{v}_d \in \mathbb{W}_d, \mathbf{v}_d = 0 \text{ on } \Gamma\} \tag{3.38}$$

and

$$L^2_d = \{q_d \mid q_d \in C^0(\Omega), q_d \in P_1,\ \forall D \in \mathbb{D}_d\} \tag{3.39}$$

where:
P_1= polynomial in three variables of degree 1.
P_2= polynomial in three variables of degree 2.

Using the above finite spaces we can approximate the weak extended formulation of the problem as:
For $t > 0$ *find* $\mathbf{u}_d(t) \in \mathbb{W}_d$, $p_d(t) \in L^2_d$, $\mathbf{U}(t) \in \mathbf{R}^3$, $\boldsymbol{\omega}(t) \in \mathbf{R}^3$ and $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}(t)$ *such that:*

$$\begin{cases} \int_\Omega \rho_F(\frac{D\mathbf{u}_d}{Dt} - \mathbf{g}) \cdot \mathbf{v}_d d\Omega - \int_\Omega p_d \nabla \cdot \mathbf{v}_d d\Omega + \int_\Omega 2\mu \mathbf{D}[\mathbf{u}_d] : \mathbf{D}[\mathbf{v}_d] d\Omega \\ + (1 - \frac{\rho_F}{\rho_s})(M_s(\frac{d\mathbf{U}}{dt} - \mathbf{g}) \cdot \widehat{\mathbf{V}} + (\mathbf{I}_s \frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\omega} \times \mathbf{I}_s \boldsymbol{\omega}) \cdot \widehat{\xi}) \\ = \langle \boldsymbol{\lambda}_d, \mathbf{v}_d - (\widehat{\mathbf{V}} + \widehat{\xi} \times \mathbf{r}) \rangle_{P(t),d} \\ \forall\, \mathbf{v}_d \in \mathbb{W}_d, \widehat{\mathbf{V}} \in \mathbf{R}^3 \text{ and } \widehat{\xi} \in \mathbf{R}^3 \\ \\ \int_\Omega q_d \nabla \cdot \mathbf{u}_d d\Omega = 0 \ \ \forall\, q_d \in L^2_d(\Omega) \\ \\ \langle \alpha_d, \mathbf{u}_d - (\mathbf{U} + \boldsymbol{\omega} \times \mathbf{r}) \rangle_{P_d(t)} = 0 \ \forall\, \alpha_d \in \boldsymbol{\Lambda}_h(t) \end{cases} \tag{3.40}$$

With initial conditions:

$$\begin{cases} \mathbf{u}_h \mid_{t=0} = \mathbf{u}_{0,d} \text{ in } \Omega \\ \int_\Omega q_d \nabla \cdot \mathbf{u}_{0,d} d\Omega = 0 \; \forall \; q_d \in L_d^2 \\ \langle \alpha_d, \mathbf{u}_{0,d} - (\mathbf{U}_0 + \boldsymbol{\omega}_0 \times \mathbf{r}) \rangle_{P_d(0)} = 0 \forall \alpha_d \text{ in } \boldsymbol{\Lambda}_d(0) \end{cases} \tag{3.41}$$

For this finite element approximation we can express the fluid velocity and the pressure at any give mesh node as a linear combination of some shape functions. Using $P_2 - P_1$ finite elements and $\boldsymbol{\phi}$ and ψ the basis functions for the velocity and pressure respectively we can express the velocity and pressure as:

$\mathbf{u}_i = \sum_{j=1}^{N_v} u_{i,j} \boldsymbol{\phi}_j$ and $p = \sum_{j=1}^{N_p} p_j \psi_j$

N_v are the velocity degrees of freedom and N_p are the pressure degrees of freedom and are determined by the type of finite element chosen for the particular discretization.

Dropping the d subscript (that denoted the discrete variables) and expanding the material derivative for the fluid velocity the system of equations given in (3.40) can be rewritten in dimensionless semi-discrete form as:

$$\begin{aligned} &\mathbf{M}_F \frac{d\mathbf{u}}{dt} + \mathbf{N}_F \mathbf{u} + \mathbf{L}_F^T p + \frac{1}{\mathrm{Re}} \mathbf{S}_F \mathbf{u} + \frac{(\rho_s - \rho_F)}{\rho_F} (M_s (\frac{d\mathbf{U}}{dt} - \mathbf{g}) \cdot \widehat{\mathbf{V}} + \mathbf{I}_s \frac{d\boldsymbol{\omega}}{dt}) \\ &= \langle \boldsymbol{\lambda}, \mathbf{v} - (\widehat{\mathbf{V}} + \widehat{\boldsymbol{\xi}} \times \mathbf{r}) \rangle_{\Omega_P(t)} \end{aligned} \tag{3.42}$$

$$\mathbf{L}_F \mathbf{u} = 0 \tag{3.43}$$

Where the matrices $\mathbf{M}_F$, $\mathbf{N}_F$, $\mathbf{S}_F$ and $\mathbf{L}_F$ are given by:

$M_{Fi,j} = \int_\Omega \boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_j d\Omega$, $S_{Fi,j} = \int_\Omega \nabla \boldsymbol{\phi}_j \cdot \nabla \boldsymbol{\phi}_i d\Omega$,

$N_{Fk,l} = \sum_{m=1}^{N_{Fv}} \mathbf{u}_k \int_\Omega \boldsymbol{\phi}_k \boldsymbol{\phi}_l \nabla \boldsymbol{\phi}_l d\Omega$, $L_{Fk,l}^i = -\int_\Omega \boldsymbol{\phi}_k \frac{d\Psi l}{dx_i} d\Omega$

3.4.2 Time Discretization

Time discretization is performed using a time-splitting method as described in Chapter 2. This approach allows the generalized Stokes and continuity equations to be treated separately from the convective terms, and these equations can be decoupled from the constraint of the rigid body motion.

Similarly to equation (2.7) the spatially discretized system can be splitted over time according to:

$$\frac{d\theta}{dt} + F_1(\theta) + F_2(\theta) + F_3(\theta) = \mathbf{f} \tag{3.44}$$

where F_1,F_2,F_3 are the three discrete spatial operators involved in the unified equation of motion.

The splitting scheme is summarized as follows:

Step1:**Convection**

Calculate the convected velocity using the method of characteristics [54]. The subproblem to be solved can be stated as:

$$\begin{aligned} &\frac{\partial \tilde{\mathbf{u}}^{n-i}(s)}{\partial s} = -(\tilde{\mathbf{u}}^{n-i}(s)\cdot\nabla)\tilde{\mathbf{u}}^{n-i}(s),\ 0 \le s \le (i+1)\Delta t,\ i = 0,1 \\ &\tilde{\mathbf{u}}^{n-i}(0) = \mathbf{u} \end{aligned} \tag{3.45}$$

where $\tilde{\mathbf{u}}$ is the convected velocity.

To solve for the convected velocity a second order accurate extrapolation of the velocity field is performed. For each point in the Eulerian mesh the boundary problem described by equation (2.16) is solved and the foot of the characteristic is determined. Using the elements containing the feet of the characteristics within the Eulerian finite element grid the velocity field at time t^{n-i} is calculated and the convected velocity field $\widetilde{\mathbf{u}}^{n-i}$ is computed. This second order scheme used to solve the convection subproblem is described in full detail in Section 2.1.1.

Step2:**Generalized Stokes**

Solve the generalized Stokes problem using the following second order scheme:

$$\frac{3\mathbf{u}^{n+\frac{1}{3}} - 4\widetilde{\mathbf{u}}^{n} + \widetilde{\mathbf{u}}^{n-1}}{2\Delta t} = -\nabla p^{n+\frac{1}{3}} + \nabla^2 \mathbf{u}^{n+\frac{1}{3}} \tag{3.46}$$

Defining: $\tau_0 = \frac{3}{2\Delta t}$, $\tau_1 = \frac{2}{\Delta t}$, $\tau_2 = \frac{1}{2\Delta t}$ and using matrix notation we can rewrite the fully discretized Stokes problem as :

$$(\tau_0 \mathbf{M}_F + \frac{1}{\mathrm{Re}}\mathbf{S}_F)\mathbf{u}^{n+\frac{1}{3}} = \mathbf{L}_F^T p^{n+\frac{1}{3}} - \tau_1 \mathbf{M}_F \widetilde{\mathbf{u}}^{n} - \tau_2 \mathbf{M}_F \widetilde{\mathbf{u}}^{n-1} \tag{3.47}$$

The generalized Stokes problem is solved using the projection method (pressure-correction method). An intermediate velocity is first calculated using a pressure at the previous step. Following, a pressure correction is computed and used to correct both the velocity and the pressure at the new time level. The pressure-correction method is described in detail in Section 2.1.2.

Step3: **Rigid body constraints**

Impose the rigid body constraints finding $\mathbf{u}$, $\mathbf{U}$, $\boldsymbol{\omega}$, $\mathbf{X}$ and $\boldsymbol{\lambda}$.

*3.1)*Compute the particle's velocity and center of mass using a second order prediction procedure:

$$\frac{3\mathbf{U}^{n+\frac{2}{3}} - 4\mathbf{U}^{n} + \mathbf{U}^{n-1}}{2\Delta t} = \frac{\mathbf{g}}{g\mathrm{Fr}} \tag{3.48}$$

$$\frac{\mathbf{X}^{n+\frac{2}{3}} - \mathbf{X}^{n-1}}{\Delta t} = \mathbf{U}^{n+\frac{2}{3}} \tag{3.49}$$

$$\frac{3\omega^{n+\frac{2}{3}} - 4\omega^{n} + \omega^{n-1}}{2\Delta t} = 0 \tag{3.50}$$

3.2) Find $\mathbf{u}^{n+1}$, $\mathbf{U}^{n+1}$, ω^{n+1}, $\mathbf{X}^{n+1}$ and $\boldsymbol{\lambda}^{n+1}$satisfying:

$$\tau_0 \mathbf{M}_F(\mathbf{u}^{n+1} - \mathbf{u}^{n+\frac{2}{3}}) + \frac{1}{\mathrm{Re}}\mathbf{S}_F\mathbf{u}^{n+1} + \tau_0(1 - \frac{\rho_s}{\rho_F})(\frac{M_s}{\rho_s}(\mathbf{U}^{n+1} - \mathbf{U}^{n+\frac{2}{3}}) \cdot \widehat{\mathbf{V}}$$
$$+ \frac{1}{\rho_L D_c}(\tau_0 \mathbf{I}_s\omega^{n+1} - \omega^{n+\frac{2}{3}} + \omega^{n+1} \times \mathbf{I}_s\omega^{n+1}) \cdot \widehat{\xi}) = \langle \boldsymbol{\lambda}^{n+1}, \mathbf{v} - (\widehat{\mathbf{V}} + \widehat{\xi} \times \mathbf{r}) \rangle_{P^{n+\frac{2}{3}}} \tag{3.51}$$

$$\langle \alpha, \mathbf{u}^{n+1} - (\mathbf{U}^{n+1} + \omega^{n+1} \times \mathbf{r}^{n+\frac{2}{3}}) \rangle_{P^{n+\frac{2}{3}}} = 0 \ \forall \alpha \in \boldsymbol{\Lambda}(t) \tag{3.52}$$

3.3) Compute the particle's center of mass using a correction procedure:

$$\frac{\mathbf{X}^{n+1} - \mathbf{X}^{n}}{\Delta t} = \frac{(\mathbf{U}^{n+1} + \mathbf{U}^{n+\frac{2}{3}})}{2} \tag{3.53}$$

Here the equations are nondimensionalized by introducing the Reynolds number of the particle, $\mathrm{Re} = \frac{U_c D_c}{\nu}$, the Froude number $\mathrm{Fr} = \frac{U_c^2}{D_c g}$,with D_c, U_c, the characteristic length and velocity of the particle, ν the kinematic viscosity and g is the magnitude of the acceleration of gravity.

Step *3.2)* is solved using a conjugate gradient algorithm presented by Glowinski et al. [32]. Equation (3.52) is actually divided into three equations and solved as follows: **Step 0: Initialization**
Assume $\boldsymbol{\lambda}^0 \in \boldsymbol{\Lambda}$. Find $\mathbf{u}^0$, $\mathbf{U}^0$, ω^0 according to:

$$(\tau_0 \mathbf{M}_F + \frac{1}{\mathrm{Re}}\mathbf{S}_F)\mathbf{u}^0 = \langle \boldsymbol{\lambda}^0, \mathbf{v} \rangle_P + \mathbf{f} \ \forall \mathbf{v} \in \mathbb{W} \tag{3.54}$$

where $\mathbf{f}$ is the contribution from the velocity at previous time levels.

$$(1 - \frac{\rho_F}{\rho_s})(\frac{M_s}{\rho_F}(\tau_0\mathbf{U}^0 + \tau_1\mathbf{U}^n + \tau_2\mathbf{U}^{n-1}) \cdot \widehat{\mathbf{V}} = -\langle \boldsymbol{\lambda}^0, \widehat{\mathbf{V}} \rangle_P \ \forall \widehat{\mathbf{V}} \in \mathbf{R}^3 \tag{3.55}$$

$$(1 - \frac{\rho_F}{\rho_s})\frac{\mathbf{I}_s}{\rho_F}(\tau_0\omega^0 + \tau_1\omega^n + \tau_2\omega^{n-1}) \cdot \widehat{\xi} = -\langle \boldsymbol{\lambda}^0, \widehat{\xi} \times \mathbf{r} \rangle_P \ \forall \widehat{\xi} \in \mathbf{R}^3 \tag{3.56}$$

Find $\boldsymbol{\lambda}_1 \in \boldsymbol{\Lambda}$ satisfying:

$$\langle \alpha, \boldsymbol{\lambda}_1 \rangle_P = \langle \alpha, \mathbf{u}^0 - (\mathbf{U}^0 + \omega^0 \times \mathbf{r}) \rangle_P \ \forall \alpha \in \boldsymbol{\Lambda} \tag{3.57}$$

Set $\boldsymbol{\lambda}_2 = \boldsymbol{\lambda}_1$. For m iteration steps assuming $\mathbf{u}^m$, $\mathbf{U}^m$, $\boldsymbol{\omega}^m$, $\boldsymbol{\lambda}^m$, $\boldsymbol{\lambda}_1^m$ and $\boldsymbol{\lambda}_2^m$ are known compute $\mathbf{u}^{m+1}$, $\mathbf{U}^{m+1}$, $\boldsymbol{\omega}^{m+1}$, $\boldsymbol{\lambda}^{m+1}$, $\boldsymbol{\lambda}_1^{m+1}$ and $\boldsymbol{\lambda}_2^{m+1}$ finding the corrections: $\tilde{\mathbf{u}}^m$, $\tilde{\mathbf{U}}^m$, $\tilde{\boldsymbol{\omega}}^m$, $\tilde{\boldsymbol{\lambda}}^m$, $\tilde{\boldsymbol{\lambda}}_1^m$ and $\tilde{\boldsymbol{\lambda}}_2^m$.

Step 1: Descent

$$(\tau_0 \mathbf{M}_F + \frac{1}{\mathrm{Re}}\mathbf{S}_F)\tilde{\mathbf{u}}^m = \langle \boldsymbol{\lambda}_2^m, \mathbf{v}\rangle_P \ \forall\, \mathbf{v} \in \mathbb{W} \tag{3.58}$$

$$(1 - \frac{\rho_F}{\rho_s})(\frac{M_s}{\rho_F}\tau_0 \tilde{\mathbf{U}}^m \cdot \widehat{\mathbf{V}}) = -\langle \boldsymbol{\lambda}_2^m, \widehat{\mathbf{V}}\rangle_P \ \forall\, \widehat{\mathbf{V}} \in \mathbf{R}^3 \tag{3.59}$$

$$(1 - \frac{\rho_F}{\rho_s})(\frac{\mathbf{I}_s}{\rho_F}\tau_0 \tilde{\boldsymbol{\omega}}^m \cdot \widehat{\boldsymbol{\xi}}) = -\langle \boldsymbol{\lambda}_2^m, \widehat{\boldsymbol{\xi}} \times \mathbf{r}\rangle_P \ \forall\, \widehat{\boldsymbol{\xi}} \in \mathbf{R}^3 \tag{3.60}$$

Find $\tilde{\boldsymbol{\lambda}}_1^m \in \boldsymbol{\Lambda}$ satisfying:

$$\langle \alpha, \tilde{\boldsymbol{\lambda}}_1^m\rangle_P = \langle \alpha, \tilde{\mathbf{u}}^m - (\tilde{\mathbf{U}}^m + \tilde{\boldsymbol{\omega}}^m \times \mathbf{r})\rangle_P \forall\, \alpha \in \boldsymbol{\Lambda} \tag{3.61}$$

Correct unknowns:

$$\beta_m = \frac{\langle \boldsymbol{\lambda}_1^m, \boldsymbol{\lambda}_1^m\rangle_P}{\langle \boldsymbol{\lambda}_2^m, \tilde{\boldsymbol{\lambda}}_1^m\rangle_P} \tag{3.62}$$

$$\boldsymbol{\lambda}^{m+1} = \boldsymbol{\lambda}^m - \beta_m \boldsymbol{\lambda}_2^m \tag{3.63}$$

$$\mathbf{u}^{m+1} = \mathbf{u}^m - \beta_m \tilde{\mathbf{u}}^m \tag{3.64}$$

$$\mathbf{U}^{m+1} = \mathbf{U}^m - \beta_m \tilde{\mathbf{U}}^m \tag{3.65}$$

$$\boldsymbol{\omega}^{m+1} = \boldsymbol{\omega}^m - \beta_m \tilde{\boldsymbol{\omega}}^m \tag{3.66}$$

$$\boldsymbol{\lambda}_1^{m+1} = \boldsymbol{\lambda}_1^m - \beta_m \boldsymbol{\lambda}_1^m \tag{3.67}$$

Step 2: New descent direction / convergence test

If

$$\frac{\langle \boldsymbol{\lambda}_1^{m+1}, \boldsymbol{\lambda}_1^{m+1}\rangle_P}{\langle \boldsymbol{\lambda}_1^0, \boldsymbol{\lambda}_1^0\rangle_P} \leq \epsilon \tag{3.68}$$

the iteration has converged and the final values for the unknown variables are those at level $m+1$. If convergence has not been achieved calculate:

$$\gamma_m = \frac{\langle \boldsymbol{\lambda}_1^{m+1}, \boldsymbol{\lambda}_1^{m+1}\rangle_P}{\langle \boldsymbol{\lambda}_1^m, \boldsymbol{\lambda}_1^m\rangle_P} \tag{3.69}$$

$$\boldsymbol{\lambda}_2^{m+1} = \boldsymbol{\lambda}_1^{m+1} + \gamma_m \boldsymbol{\lambda}_2^m \tag{3.70}$$

Set $m = m+1$ and go to **Step 1**.

3.5 Code validation

Analytical solutions for flow around and transfer from rigid and fluid spheres are effectively limited to Reynolds numbers smaller than 1. In the absence of analytic results, information is obtained from experimental observations, numerical solutions and boundary-layer approximations [17]. Clift el al. [17] studied the development of the flow field with the Reynolds number for a fluid flowing past a rigid sphere. They classified the flow into six different regions:

1. **Unseparated flow** ($1 < \mathbf{Re} < 20$) In this region the flow is axisymmetric and perfectly laminar.

2. **Onset of the separation** ($\mathbf{Re} = 20$) Flow separation occurs. The exact Reynolds number where the separation occurs has been questioned and debated by many authors (see [68], [58], [52], [46], [22],[72]). Many numerical solutions as well as the drag determination indicate a change in the flow regime at Reynolds = 20 [17].

3. **Steady wake region** ($20 < \mathbf{Re} < 130$) In this region the separation moves forward so that the attached recirculation wake widens and lengthens. Taneda [68] showed the development of the wake for flow past a rigid sphere.

4. **Onset of the wake instability** ($130 < \mathbf{Re} < 400$) As the Reynolds number increases diffusion and convection of vorticity no longer keep pace with vorticity generation. The Reynolds at which vorticity shedding begins is called the "lower critical Reynolds number". At Reynolds of 130 a weak long period oscillation appears in the tip of the wake. For Reynolds up to 200 the flow is steady and axisymmetric ([44], [72]). Natarajan and Acrivos [59] studied the stability of this axisymmetric flow. They reported a bifurcation at Re=210 and suggested this corresponded to the transition from steady axisymmetric to the steady non-axisymmetric double-thread wake. At about Re = 270 the large vortices associated with pulsation of the fluid circulating in the wake, periodically form and move downstream. Vortex shedding appears to result from flow instabilities originating in the free surface layer and moving downstream to affect the position of the wake tip [63]. A number of researchers including Achenbach [1], Kim and Durbin [47] and Sakamoto and Haniu [64] have presented measurements of the frequencies present in the unsteady wake.

5. **High sub-critical Reynolds number** ($400 < \mathbf{Re} < 3.5\ 10^5$) As Reynolds increases beyond about 400 vortices are shed as a regular succession of loops from alternate sides of a plane while processes slowly about the axis. Shed loops progressively lose their character and may combine to form vortex balls. By Re = 1300 the wake shows three dimensional rotation. Separation occurs at a point that rotates around the sphere at the shedding frequency. The wake may appear as a pair of vortex filaments [17].

6. **Critical transition and supercritical flow** ($\mathbf{Re} > 3.5\ 10^5$) Beyond this Reynolds number changes in the flow pattern occur. The separation point moves and the fluctuation in the position become more marked. The detached free surface layer becomes turbulent soon after separation [17].

The code was tested with various flows involving a sedimenting single particle. The cases were set for Reynolds numbers smaller that 400. Our results were compared to those presented by Mordant et al. [57] from their experimental work. In all cases the difference between the numerical results and the experimental value was not larger than 17% (depending on the mesh resolution).

The first result that we show here is for the case of a sphere of radius = 0.5 settling in a rectangular channel. The channel's width is 4 times the diameter of the sphere. The dimensions of the channel are ($4 \times 4 \times 12$). The fluid's mesh has 179401 nodes and 120000 elements. The sphere's mesh has 76 nodes and 226 elements. Figures 3.2 and 3.3 show the discretized domain.

The Reynolds number is set to 21.0 and the Froude number is 2.24. The timestep is 0.001. The sphere settles under gravity in a fluid at rest. The boundary conditions for the channel are zero velocity for the nodes on the walls. The ratio of densities between the sphere and the fluid is set to 2.56.

Figure 3.4 to 3.6 show the result of this simulation. Figure 3.4 shows the sphere's position at different timesteps. The flow pattern is shown in Figures 3.5 and 3.6. In the first case the streamlines corresponding to the absolute motion velocity field are plotted. In the second case the velocity field is relative to the sphere.

Figure 3.7 shows the sphere's sedimentation velocity with time. The particle accelerates immediately after it is released but eventually reaches an approximately constant sedimentation velocity when gravity balances buoyancy and the drag force. In this figure the velocity obtained with the simulation is compared to the curve proposed by Mordant et al. [57]. These authors performed experimental studies of

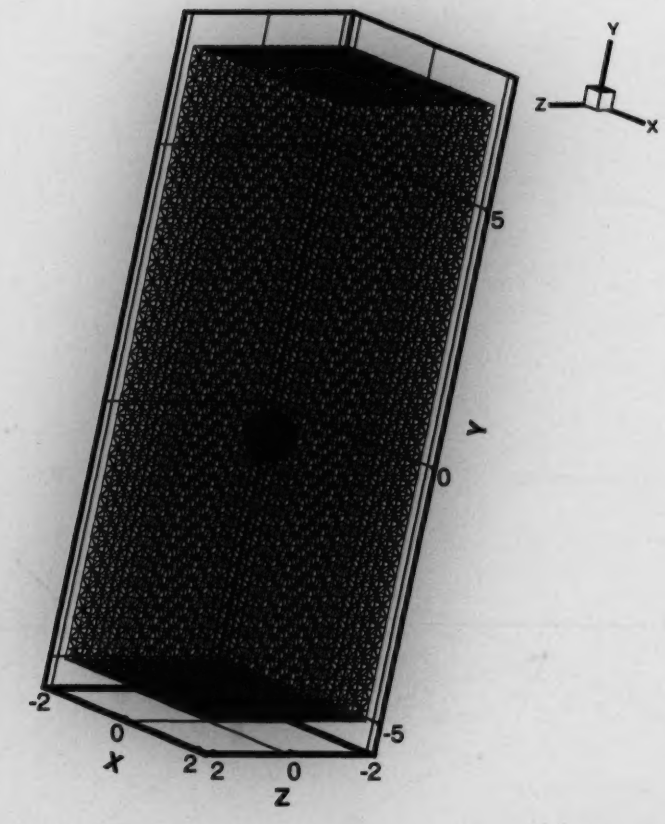

Figure 3.2: Computational domain for one sphere in a channel.

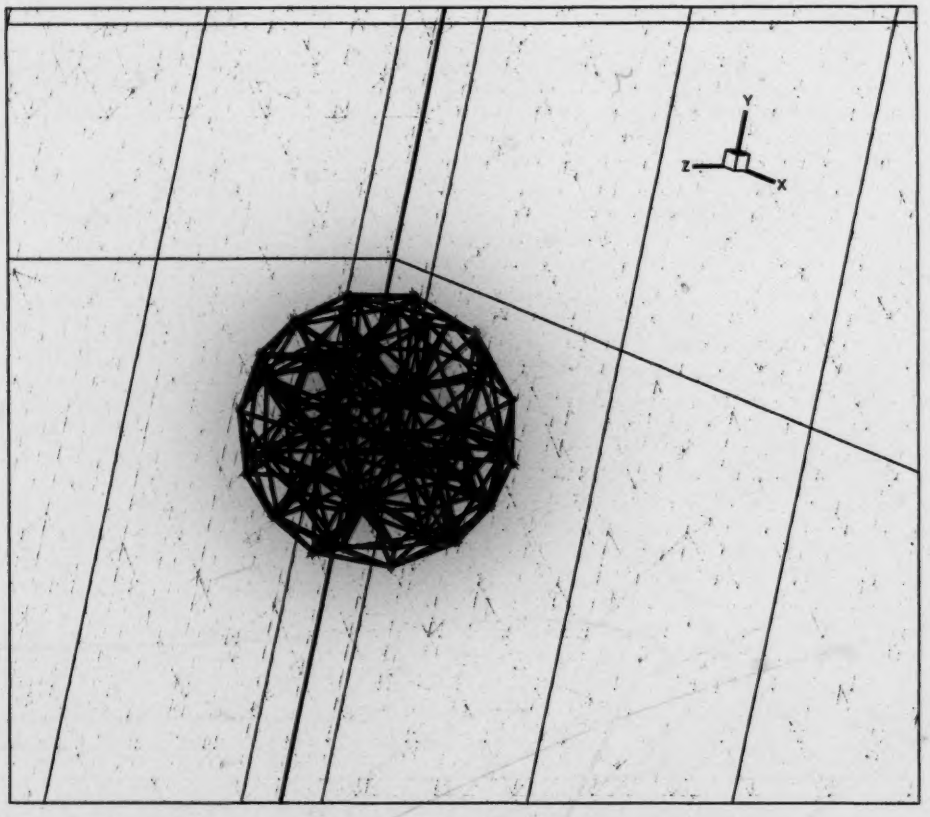

Figure 3.3: Computational domain for one sphere.

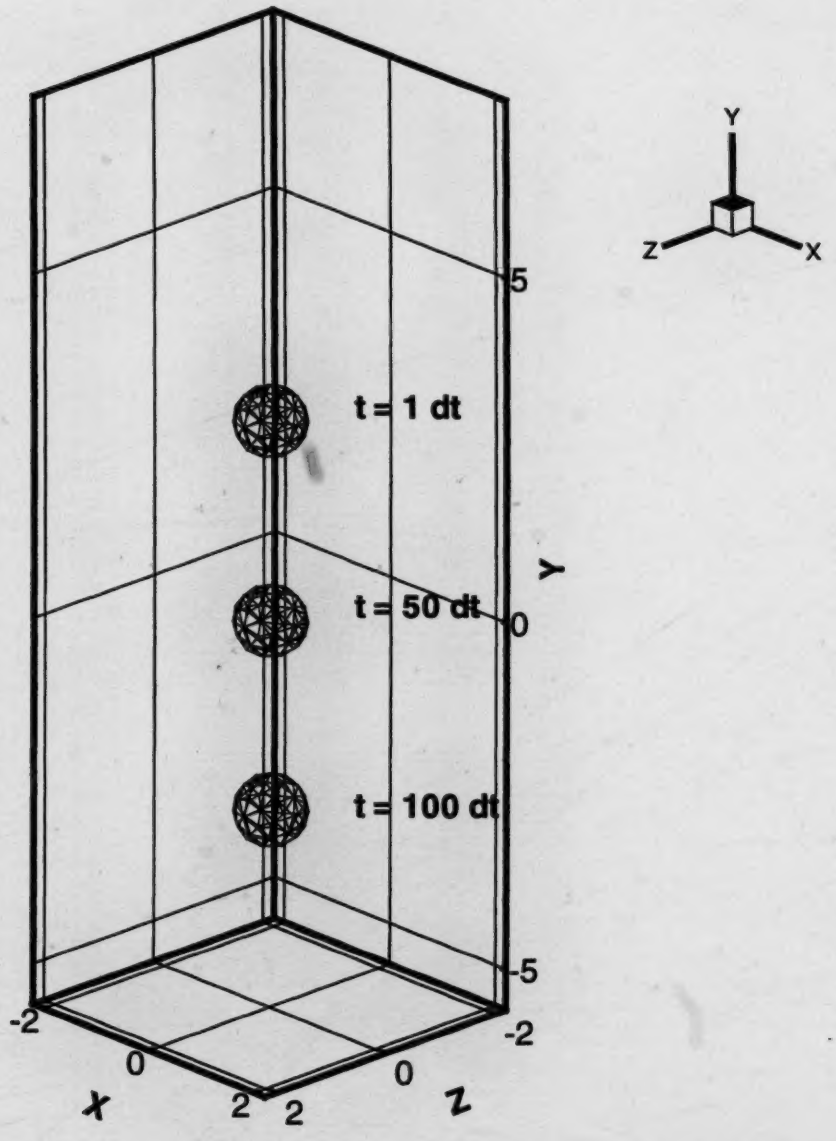

Figure 3.4: Sphere's position at different times. Re=21.0 Fr=2.24.

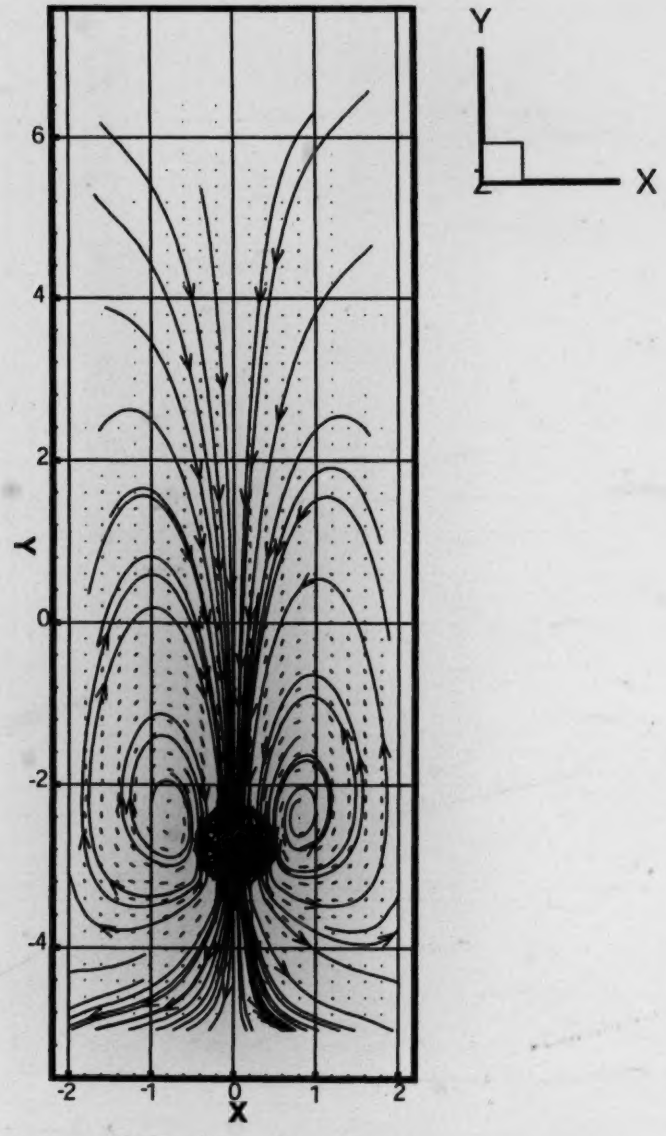

Figure 3.5: Sphere settling in a channel, streamlines.

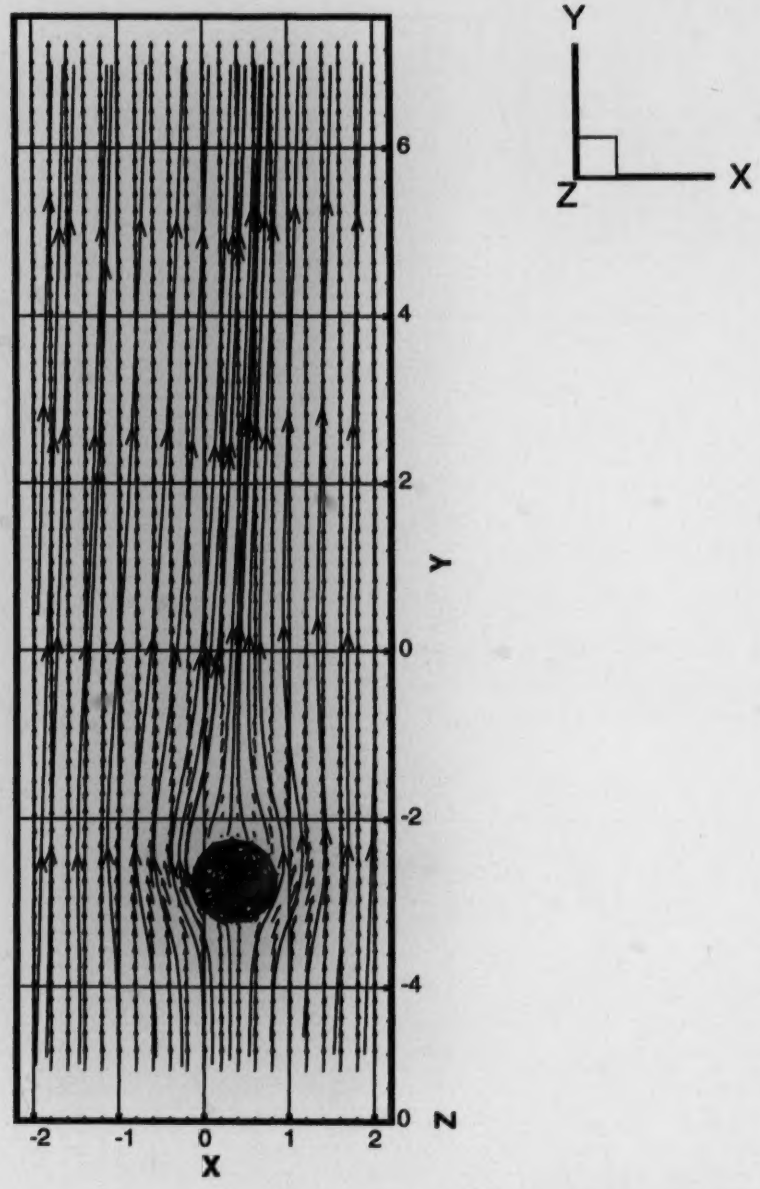

Figure 3.6: Sphere settling in a channel, relative motion.

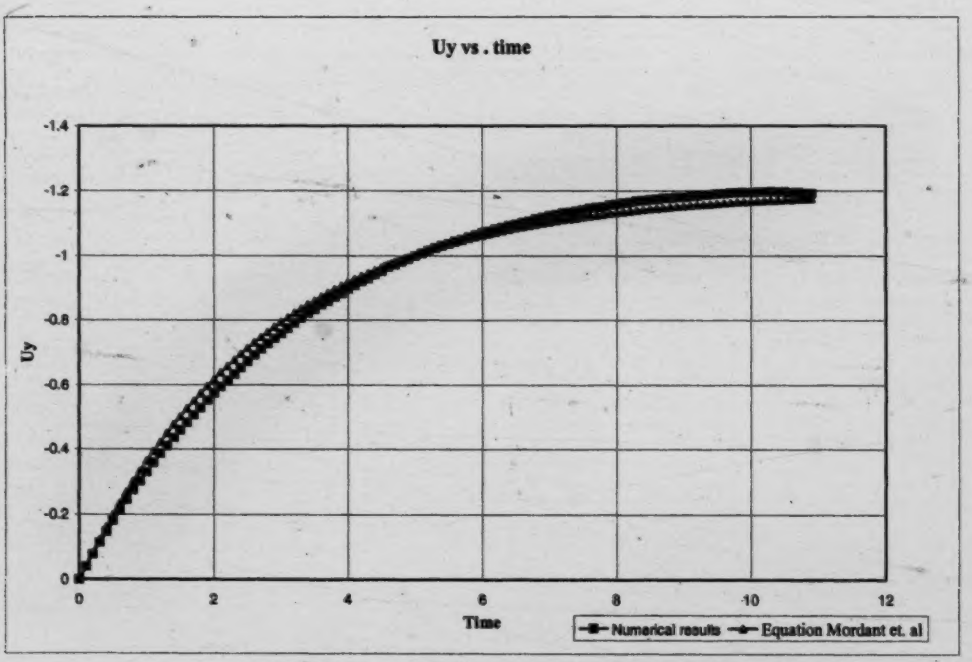

Figure 3.7: Particle's vertical velocity vs. time (dimensionless).

the motion of a solid sphere settling under gravity in a fluid at rest. All their curves collapsed onto a single exponential shape given by:

$$\frac{v_{\text{exp}}}{V_{\text{Terminal}}} = 1 - e^{(-\frac{3t}{\tau_{95}})}$$

where v_{exp} is the vertical velocity of the particle, V_{Terminal} is the terminal velocity, t is the time, and τ_{95} is the time it takes the particle to reach 95% of the limit speed.

The shape of our simulation curve matches the one produced using the equation presented by these authors. The value of the simulation terminal velocity is higher (about 15%) than the experimental one. This discrepancy could be attributed to the fact that our particle is a linear approximation to a sphere since we use linear (flat) elements to enforce the boundary conditions. This limitation could be overcome using an algorithm with a second order approximation for the Lagrange multipliers and isoparametric elements so that spherical particles can be exactly discretized.

For a solid sphere moving in a viscous fluid with a constant velocity at moderate Reynolds number a recirculation area or wake can be observed. This wake is formed due to a positive pressure gradient and can be understood by examining the nature of fluids. All fluids naturally flow from regions of high pressure to low pressure (in the direction of a negative pressure gradient). When the flow approaches the point where it must turn to go around the maximum cross-section of the sphere a positive pressure gradient forms due to the increase in space around the sphere (this positive pressure gradient is commonly called an adverse pressure gradient). The positive pressure gradient causes the fluid to reverse direction and form a wake. Taneda [68] presented photographs of fluid around a moving sphere for Reynolds number between 9.15 and 133. The flow pattern was visualized using aluminum dust. In his experimental work Taneda analyzed the relation between the size of the vortex ring behind the sphere and the Reynolds number. He established that for Reynolds number lower than 22 the flow around the sphere is perfectly laminar and there is no wake formed. When Reynolds number is about 25 a small permanent vortex ring is formed in the neighborhood of the rear stagnation point. As the Reynolds number increases this wake grows and becomes more elongated in the flow direction. Taneda's experimental work shows that the vortex-ring is nearly proportional to the logarithm of the Reynolds number. He also determined the center of the vortex-ring for Reynolds numbers larger than 50. The flow behind the vortex-ring was found to be perfectly laminar until a Reynolds number of about 200 was reached. The front-part of the vortex-ring was attached to the sphere up to a Reynolds number of 300

or so [68]. Nakamura [58] studied the characteristics of the steady wake behind a sphere. His work suggested the possibility of the existence of a closed recirculation eddy behind a sphere even at Reynolds number as low as 10. His experimental work also showed that the eddy could preserve up to the case of Reynolds of 190 (based on the diameter of the sphere). He also presented a new profile for the variation of the wake separation angles against the steady state Reynolds number. In our simulations no wake was produced for Reynolds numbers smaller than 20.

Figures 3.8 and 3.9 show the streamlines for the case of a sphere settling with a higher Reynolds number (Re=183.0 and Fr=3.75). At this Reynolds number a pair of vortices can be clearly seen behind the sphere.

These results agree to the results shown by Taneda ([68],[69]).

3.6 Conclusions

The goal of this study is to develop an efficient parallel algorithm for computing the dynamics of a large number of solid particles in a liquid flow.

As a first approach an algorithm that applies a fictitious domain/ distributed Lagrange multipliers method to solve for the incompressible Navier-Stokes combined with rigid solid particles equations was developed. The code was tested by comparing the results for a sedimenting sphere in different physical settings to some available experimental and numerical results. The present algorithm was implemented for the case of a single solid particle. The code can be extended to take into account many particles but a suitable collision detection mechanism would have to be developed.

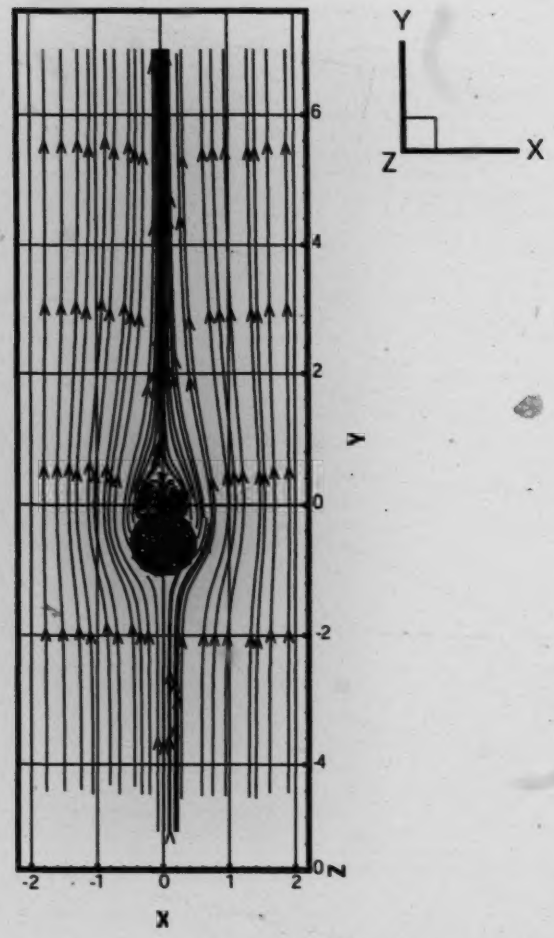

Figure 3.8: Streamlines for Re=183.0 Fr=3.75.

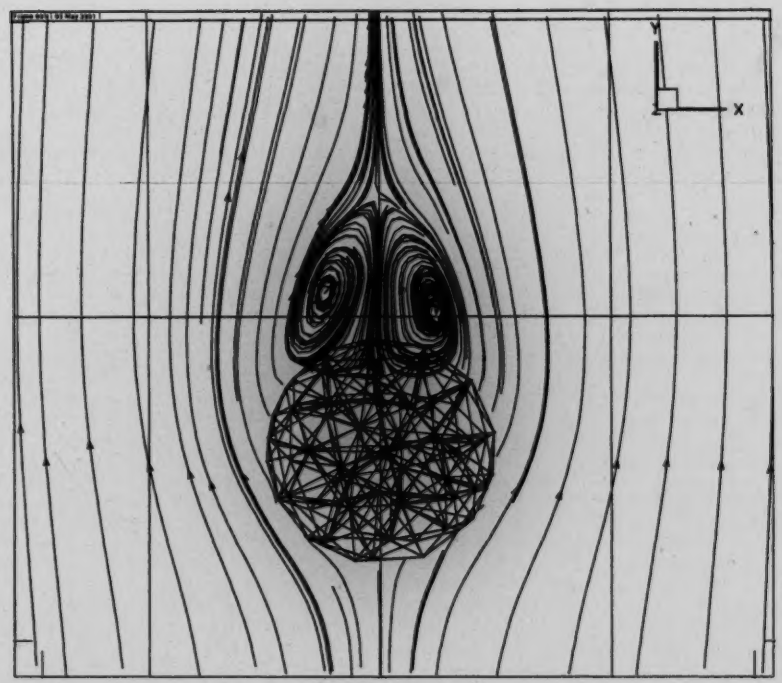

Figure 3.9: Streamlines for Re=183.0 Fr=3.75. Wake formation.

Chapter 4

A Fictitious domain approach using global Lagrange multipliers

The Glowinski's distributed Lagrange multipliers approach [33] requires the use of an unstructured grid over each particle, and subsequently solving at each iteration level the linear system for each of the particles. This is quite inconvenient algorithmically and can be computationally inefficient, particularly in the case of many particles. Therefore, we suggested the following global Lagrange multiplier approach.

4.1 Starting formulation

We consider a bounded domain Ω_1 with an external boundary Γ filled with a Newtonian liquid with density ρ_1 and viscosity μ_1. Within this liquid we consider n rigid particles occupying a domain $\Omega_2 = \cup_{i=1}^{n} \Omega_{2,i}$ and having densities $\rho_{2,i}, i = 1, \ldots, n$. Let us also denote the interface between Ω_1 and Ω_2 by Σ. The equations of motion of the fluid in Ω_1 are the Navier-Stokes equations (presented here in a dimensionless, stress-divergence form)

$$\rho_1 \frac{D\hat{\mathbf{u}}_1}{Dt} = \nabla \cdot \hat{\boldsymbol{\sigma}}_1, \quad \nabla \cdot \hat{\mathbf{u}}_1 = 0 \text{ in } \Omega_1 \tag{4.1}$$

where $\hat{\mathbf{u}}_1$ is the velocity, D/Dt denotes the full derivative in time (including the advection terms), $\hat{\boldsymbol{\sigma}}_1$ is the stress tensor, and ρ_1 is the density. $\hat{\boldsymbol{\sigma}}_1$ is defined as usual $\hat{\boldsymbol{\sigma}}_1 = \hat{p}_1 \boldsymbol{\delta} + 2\mu_1 \mathbb{D}[\hat{\mathbf{u}}_1]$ with $\hat{p}_1$ being the pressure in the liquid phase, $\mathbb{D}[\hat{\mathbf{u}}_1] = 0.5[\nabla \hat{\mathbf{u}}_1 + (\nabla \hat{\mathbf{u}}_1)^T]$ being the rate-of-strain tensor, and $\boldsymbol{\delta}$ being the Kronecker tensor. The boundary conditions on Γ are not of a concern for the present method and

therefore we assume the simplest case of homogeneous Dirichlet conditions. On the internal boundary of Ω_1, Σ, we presume no slip condition for the velocity. The problem requires initial condition for the velocity which we presume to be in the form $\hat{\mathbf{u}}_1(\mathbf{x},0) = \mathbf{u}_0, \quad \nabla\cdot\mathbf{u}_0 = 0$. The equations of motion of a rigid particle are usually written in terms of the velocity of the centroid and the angular velocity , six equations in 3D. For the purpose of the present method, however, it is more convenient to define a velocity field $\mathbf{u}_2(\mathbf{x},t)$ which is continuous within the region occupied by any given particle, and zero in Ω_1. Clearly, $\mathbf{u}_2 \in \mathbb{L}^2(\Omega)$. If $\mathbf{U}_i(t), \boldsymbol{\omega}_i(t), \mathbf{X}_i$ are the velocity of the centroid, the angular velocity and the coordinate vector of the ith particle then $\mathbf{u}_2(\mathbf{x},t)|_{\Omega_i} = \mathbf{U}_i(t) + \boldsymbol{\omega}_i(t) \times (\mathbf{x} - \mathbf{X}_i(t))$. The restriction of the momentum equation for $\mathbf{u}_2$ on $\Omega_{2,i}$, $\mathbf{u}_{2,i}$ can be written in the following integral form

$$\frac{d}{dt}\int_{\Omega_{2,i}} \rho_{2,i}\mathbf{u}_2 d\Omega = \int_{\Omega_{2,i}} \rho_{2,i}\mathbf{g} d\Omega + \int_{\partial\Omega_{2,i}} \hat{\boldsymbol{\sigma}}_1 \mathbf{n}_i ds \tag{4.2}$$

with $\mathbf{n}_i$ being the outward normal to the surface of the ith particle. The last term in this equation represents the total hydrodynamic force acting on the surface of the ith particle. Let us denote by $\boldsymbol{\sigma}_1$ the continuous extension of the stress $\hat{\boldsymbol{\sigma}}_1$ over the entire domain Ω. Such an extension can always be constructed and this is the basis for the different fictitious domain methods that are proposed in the literature. Then using the divergence theorem we can rewrite (4.2) in the following form

$$\int_{\Omega_{2,i}} \frac{D}{Dt}(\rho_{2,i}\mathbf{u}_2)\, d\Omega = \int_{\Omega_{2,i}} \rho_{2,i}\mathbf{g} d\Omega + \int_{\Omega_{2,i}} \nabla\cdot\boldsymbol{\sigma}_1 d\Omega. \tag{4.3}$$

A natural way to extend the stress tensor in $\Omega_{2,i}$ is to assume that it is Newtonian and extend the velocity field $\hat{\mathbf{u}}_1$ to $\mathbf{u}_1$ such that $\mathbf{u}_1|_{\Omega_1} = \hat{\mathbf{u}}_1$, the pressure to some p_1 such that $p_1|_{\Omega_1} = \hat{p}_1$, and write $\nabla\cdot\boldsymbol{\sigma}_1 = -\nabla p_1 + \mu_1\nabla^2\mathbf{u}_1$. Then (4.3) can be rewritten as

$$\int_{\Omega_{2,i}} \frac{D}{Dt}(\rho_{2,i}\mathbf{u}_{2,i})\, d\Omega = \int_{\Omega_{2,i}} \rho_{2,i}\mathbf{g} d\Omega + \int_{\Omega_{2,i}} \left(-\nabla p_1 + \mu_1\nabla^2\mathbf{u}_1\right) d\Omega. \tag{4.4}$$

Now if we adopt the notation

$$\mathbf{F} = \begin{cases} -\rho_1\frac{D\mathbf{u}_1}{Dt} + \mu_1\nabla^2\mathbf{u}_1 - \nabla p_1, & \text{in} \quad \Omega_{2,i}, \quad i = 1,\ldots,n \\ 0, \quad \text{in} \quad \Omega_1 \end{cases} \tag{4.5}$$

the equation for the ith particle momentum becomes

$$\int_{\Omega_{2,i}} \frac{D}{Dt}(\rho_{2,i}\mathbf{u}_2 - \rho_1\mathbf{u}_1)\, d\Omega = \int_{\Omega_{2,i}} (\rho_{2,i}\mathbf{g} + \mathbf{F}) d\Omega. \tag{4.6}$$

Because of (4.5) we can now extend the momentum equation (4.1) to the entire Ω as

$$\rho_1 \frac{D\mathbf{u}_1}{Dt} = -\nabla p_1 + \mu_1 \nabla^2 \mathbf{u}_1 - \mathbf{F}, \quad \nabla \cdot \mathbf{u}_1 = 0 \text{ in } \Omega. \tag{4.7}$$

The additional force per unit volume $\mathbf{F}$ in this equation can be interpreted as the interaction force between the two phases. As discussed by Glowinski et al. [33] it enforces the rigid body motion onto the fluid velocity field within each of the particles. If we adopt this additional constraint onto $\mathbf{u}_1$ i.e.

$$\mathbf{u}_1 = \mathbf{u}_2 \text{ in } \Omega_{2,i} \tag{4.8}$$

then equation of momentum of the ith particle, (4.6), can be written as

$$\int_{\Omega_{2,i}} \frac{D}{Dt} \left[(\rho_{2,i} - \rho_1)\mathbf{u}_2 \right] d\Omega = \int_{\Omega_{2,i}} (\rho_{2,i}\mathbf{g} + \mathbf{F}) d\Omega. \tag{4.9}$$

Now we recall that $\mathbf{u}_2$ is a rigid body velocity field i.e. $\mathbf{u}_2(\mathbf{x}, t)|_{\Omega_i} = \mathbf{U}_i(t) + \boldsymbol{\omega}_i(t) \times (\mathbf{x} - \mathbf{X}_i(t))$. Then (4.9) implies the following equation for $\mathbf{U}_i$

$$\Delta M_i \frac{d\mathbf{U}_i}{dt} = M_i \mathbf{g} + \int_{\Omega_{2,i}} \mathbf{F} d\Omega. \tag{4.10}$$

where $M_i = \int_{\Omega_{2,i}} \rho_{2,i} d\Omega, \Delta M_i = \int_{\Omega_{2,i}} (\rho_{2,i} - \rho_1) d\Omega$. The angular velocity $\boldsymbol{\omega}_i$ can be recovered from the no-slip boundary condition on the surface of the ith particle. It reads

$$\mathbf{U}_i(t) + \boldsymbol{\omega}_i(t) \times (\mathbf{x} - \mathbf{X}_i(t)) = \mathbf{u}_1, \text{ on } \partial\Omega_{2,i}.$$

Then clearly

$$\int_{\partial\Omega_{2,i}} [(\boldsymbol{\omega}_i(t) \times (\mathbf{x} - \mathbf{X}_i(t))] \times \mathbf{n} ds = \int_{\partial\Omega_{2,i}} (\mathbf{u}_1 - \mathbf{U}_i(t)) \times \mathbf{n} ds$$

which yields, using the Stokes theorem, that

$$\boldsymbol{\omega}_i(t) V_{\Omega_i} = \int_{\Omega_{2,i}} \nabla \times (\mathbf{u}_1 - \mathbf{U}_i(t)) d\Omega,$$

where V_{Ω_i} is the volume of the ith particle. Finally, the set of equations of the fictitious domain method is given by

$$\rho_1 \frac{D\mathbf{u}_1}{Dt} = -\nabla p_1 + \mu_1 \nabla^2 \mathbf{u}_1 - \mathbf{F}, \quad \nabla \cdot \mathbf{u}_1 = 0 \text{ in } \Omega \tag{4.11}$$

$$\Delta M_i \frac{d\mathbf{U}_i}{dt} = M_i \mathbf{g} + \int_{\Omega_{2,i}} \mathbf{F} d\Omega, \quad i = 1, \ldots, n \tag{4.12}$$

$$\boldsymbol{\omega}_i(t) V_{\Omega_i} = \int_{\Omega_{2,i}} \nabla \times (\mathbf{u}_1 - \mathbf{U}_i(t)) d\Omega, \quad i = 1, \ldots, n \tag{4.13}$$

$$\mathbf{U}_i(t) + \boldsymbol{\omega}_i(t) \times (\mathbf{x} - \mathbf{X}_i(t)) = \mathbf{u}_1, \text{ in } \Omega_{2,i}, \quad i = 1, \ldots, n. \tag{4.14}$$

From this set of equations, it is clear that $\mathbf{F}$ is a Lagrange multiplier that enforces the constraint (4.14) onto $\mathbf{u}_1$ and $\mathbf{u}_2$. It is non zero only within the domain occupied by the particles and therefore it is closely related to the distributed Lagrange multiplier of Glowinski et al. [33]. However, its impact on the fluid velocity field is global, over the entire domain, due to the equation (4.11). This led us to the idea of defining another global Lagrange multiplier, $\boldsymbol{\lambda}$, which is related to $\mathbf{F}$ through the following boundary value problem

$$\begin{aligned} -\alpha\boldsymbol{\lambda} + \mu_1\nabla^2\boldsymbol{\lambda} &= \mathbf{F}, \text{ in } \Omega \\ \boldsymbol{\lambda} &= 0, \text{ on } \Gamma, \end{aligned} \tag{4.15}$$

where $\alpha > 0$ is a constant to be defined later. This is a well posed problem for $\mathbf{F} \in \mathbb{L}^2$, and it is more convenient to use its unique solution to impose the constraint (4.14). The benefit of using it, is that it has the same regularity as $\mathbf{u}_1$ and therefore they can be discretized on the same grid (i.e. discretization space). It is very similar to the distributed Lagrange multiplier $\boldsymbol{\lambda}$ of Glowinski et al. [33] provided that the constraint is imposed using an H^1 inner product. However, with the present definition of $\boldsymbol{\lambda}$ and a proper choice for α, it will become clear in the next section, that we can significantly decrease the computational expenses for the iterative solution of the system (4.11)-(4.14).

Substituting $\mathbf{F}$ from (4.15) into (4.11)-(4.14) we obtain the following system which will be used further to compute $\mathbf{u}_1$, $\mathbf{U}_i$ and $\boldsymbol{\omega}_i$

$$\rho_1\frac{D\mathbf{u}_1}{Dt} = -\nabla p_1 + \mu_1\nabla^2\mathbf{u}_1 + \alpha\boldsymbol{\lambda} - \mu_1\nabla^2\boldsymbol{\lambda}, \quad \nabla\cdot\mathbf{u}_1 = 0 \text{ in } \Omega \tag{4.16}$$

$$\Delta M_i\frac{d\mathbf{U}_i}{dt} = M_i\mathbf{g} - \int_{\Omega_{2,i}} \alpha\boldsymbol{\lambda}d\Omega - \mu_1\int_{\partial\Omega_{2,i}} \frac{\partial\boldsymbol{\lambda}}{\partial\mathbf{n}}ds, \quad i = 1,\ldots,n \tag{4.17}$$

$$V_{\Omega_i}\boldsymbol{\omega}_i = \int_{\Omega_{2,i}} \nabla\times(\mathbf{u}_1 - \mathbf{U}_i)d\Omega, \quad i = 1,\ldots,n \tag{4.18}$$

$$\mathbf{U}_i(t) + \boldsymbol{\omega}_i(t)\times(\mathbf{x} - \mathbf{X}_i(t)) = \mathbf{u}_1, \text{ in } \Omega_{2,i}, \quad i = 1,\ldots,n. \tag{4.19}$$

Note that in the second equation we have integrated $\nabla^2\boldsymbol{\lambda}$ by parts. Note also that this choice for $\boldsymbol{\lambda}$ is similar to the choice of Lagrange multiplier in [33] if the inner product for imposition of the rigid body constraint is chosen to be the H^1 inner product. In the present case, however, $\boldsymbol{\lambda}$ is defined over the entire computational domain while in [33] it is defined only locally, within the areas occupied by the particles. The global Lagrange multiplier will allow for a change in the fictitious domain method as proposed in [33], which generally increases the efficiency of the method and simplifies it (see the next section).

In addition to the system (4.16)-(4.19) we also need to solve an equation for the position of the centre of mass of each particle which reads

$$\frac{\partial \mathbf{X}_i}{\partial t} = \mathbf{U}_i, \quad i = 1, \ldots, n.$$

4.2 Discretization procedure

We discretize the system (4.16)-(4.19) using a finite element procedure although finite difference or finite volume based procedures can also be used. Note that only (4.16) and (4.19) require weak formulations. Equations (4.17) and (4.18) are an ordinary and an algebraic equation. The overall problem is in fact a Navier-Stokes problem with an additional constraint (4.19), and therefore we chose to discretize it with a projection scheme (for $\mathbf{u}_1$ and p_1) combined with an additional iteration for the imposition of the rigid body constraint. The scheme is derived from the (formally) second order in time characteristic/projection scheme described in [56]. In space, we used $\mathbb{P}_2 - P_1$ finite elements which results in a (formally) second order spatial discretization. The algorithm can be summarized in the following four steps.

Substep 1 (advection).

The advective part of the system is integrated with the method of characteristics [56]. If $\underline{\mathbf{x}}$ is an approximation of the foot of the characteristic originating at $\mathbf{x}$ then the advected velocity is given by $\tilde{\mathbf{u}}_1^n(\mathbf{x}) = \mathbf{u}_1^n(\underline{\mathbf{x}}), \tilde{\mathbf{u}}_1^{n-1}(\mathbf{x}) = \mathbf{u}_1^{n-1}(\underline{\mathbf{x}})$. $\underline{\mathbf{x}}$ is usually approximated with an Euler explicit scheme [56].

The center of mass of the ith particle is predicted explicitly as

$$\mathbf{X}_i^{p,n+1} = \mathbf{X}_i^{n-1} + 2\delta t \mathbf{U}_i^n \tag{4.20}$$

where δt is the time step.

Substep 2 (diffusion).

If we set $\tau_0 = 3/(2\delta t), \tau_1 = -2\delta t, \tau_2 = 1/(2\delta t)$ then this substep can be written as

$$\begin{aligned} &\rho_1 \tau_0 \mathbf{u}_1^* - \mu_1 \nabla^2 \mathbf{u}_1^* = -\rho_1(\tau_1 \tilde{\mathbf{u}}_1^n - \tau_2 \tilde{\mathbf{u}}_1^{n-1}) - \nabla p^n, \text{ in } \Omega \\ &\mathbf{u}_1^* = 0 \text{ on } \Gamma \end{aligned} \tag{4.21}$$

Substep 3 (incompressibility)

$$\begin{aligned}
&\tau_0(\mathbf{u}_1^{**} - \mathbf{u}_1^{*}) = -\nabla(p_1^{n+1} - p_1^{n}) \text{ in } \Omega \\
&\nabla\cdot\mathbf{u}_1^{**} = 0 \text{ in } \Omega \\
&\mathbf{u}_1^{**}\cdot\mathbf{n} = 0 \text{ on } \Gamma,
\end{aligned} \tag{4.22}$$

$\mathbf{n}$ being the outward normal to Γ.

Substep 4 (rigid body constraint)

The rigid body motion is imposed iteratively using the following iteration. Let us first set the 0-th approximation for $\boldsymbol{\lambda}^{n+1}, \mathbf{u}_1^{n+1}, \mathbf{U}_i^{n+1}, \boldsymbol{\omega}_i^{n+1}$ and $\mathbf{u}_2^{n+1}$ as

$$\begin{aligned}
&\boldsymbol{\lambda}^{0,n+1} = 0 \\
&\mathbf{u}_1^{0,n+1} = \mathbf{u}_1^{**} \\
&\tau_0\mathbf{U}_i^{0,n+1} = -\tau_1\mathbf{U}_i^{n} - \tau_2\mathbf{U}_i^{n-1} + \frac{M_i}{\Delta M_i}\mathbf{g} \\
&V_{\Omega_i}\boldsymbol{\omega}_i^{0,n+1} = \int_{\Omega_i} \nabla\times\mathbf{u}_1^{0,n+1} d\Omega \\
&\mathbf{u}_2^{0,n+1} = \mathbf{U}_i^{0,n+1} + \boldsymbol{\omega}^{0,n+1}\times(\mathbf{x} - \mathbf{X}_i^{p,n+1}).
\end{aligned}$$

We also need to set $\boldsymbol{\lambda}^{1,n+1} = \boldsymbol{\lambda}^{n}$. If we denote the difference between two subsequent iterations for a quantity Q by $\delta_t Q$ i.e. $\delta_t Q^{k+1} = Q^{k+1} - Q^{k}$ then the subsequent iterates are computed for $k \geq 0$ by

$$\begin{cases}
(\tau_0 I - \mu_1\nabla^2)\delta_t\mathbf{u}_1^{k+1,n+1} = (\alpha I - \mu_1\nabla^2)\delta_t\boldsymbol{\lambda}^{k+1,n+1} & \text{in } \Omega \\
\delta_t\mathbf{u}_1^{k+1,n+1} = 0 & \text{on } \Gamma
\end{cases} \tag{4.23}$$

$$\begin{cases}
\Delta M_i\tau_0\delta_t\mathbf{U}_i^{k+1,n+1} = -\int_{\Omega_{2,i}} \alpha\delta_t\boldsymbol{\lambda}^{k+1,n+1} d\Omega & \\
\quad +\mu_1\int_{\partial\Omega_{2,i}} \frac{\partial\delta_t\boldsymbol{\lambda}^{k+1,n+1}}{\partial\mathbf{n}} ds & i = 1,\dots,n \\
V_{\Omega_i}\boldsymbol{\omega}_i^{k+1,n+1} = \int_{\Omega_i}\nabla\times\mathbf{u}_1^{k+1,n+1} d\Omega & i = 1,\dots,n \\
\mathbf{u}_2^{k+1,n+1} = \mathbf{U}_i^{k+1,n+1} + \boldsymbol{\omega}_i^{k+1,n+1}\times(\mathbf{x} - \mathbf{X}_i^{p,n+1}) & \text{in } \Omega_{2,i} \quad i = 1,\dots,n \\
\mathbf{u}_2^{k+1,n+1} = 0 & \text{in } \Omega_1
\end{cases} \tag{4.24}$$

$$\begin{cases}
(1 + \frac{\rho_1}{\rho_{2,i} - \rho_1})(\alpha I - \mu_1\nabla^2)\delta_t\boldsymbol{\lambda}^{k+2,n+1} & \\
= (\rho_1\tau_0 I - \mu_1\nabla^2)(\delta_t\mathbf{u}_1^{k+1,n+1} - \delta_t\mathbf{u}_2^{k+1,n+1}) & \text{in } \Omega_{2,i}, \quad i = 1,\dots,n \\
(1 + \frac{\rho_1}{\rho_{2,i} - \rho_1})(\alpha I - \mu_1\nabla^2)\delta_t\boldsymbol{\lambda}^{k+2,n+1} = 0 & \text{in } \Omega_1 \\
\delta_t\boldsymbol{\lambda}^{k+2,n+1} = 0 & \text{on } \Gamma,
\end{cases} \tag{4.25}$$

where I is the identity operator. Note that in the last set of equations that determines the increment of $\boldsymbol{\lambda}$ we used the fact that $\mathbf{u}_2$ is a rigid body velocity field inside each particle and therefore $\nabla^2\mathbf{u}_2 = 0$ in $\Omega_{2,i}$.

Upon convergence for some $k = N$ we set $\mathbf{u}_1^{n+1} = \mathbf{u}_1^{N+1,n+1}, \mathbf{U}_i^{n+1} = \mathbf{U}_i^{N+1,n+1}$, $\boldsymbol{\omega}^{n+1} = \boldsymbol{\omega}^{N+1,n+1}$, $\boldsymbol{\lambda}^{n+1} = \boldsymbol{\lambda}^{N+1,n+1}$. The equation of the center of mass of the particles is solved with a second order predictor-corrector procedure the predictor being given by (4.20) and the corrector being given by

$$\mathbf{X}_i^{n+1} = \mathbf{X}_i^{p,n+1} + 0.5\delta t(\mathbf{U}_i^{n+1} + \mathbf{U}_i^n) \tag{4.26}$$

Similarly to the iteration in [33], instead of the present (Richardson) iteration we can also adopt a conjugate gradient type of iterative algorithm for imposition of the rigid body constraint. The numerical tests show that only few iterations of the present procedure are necessary per time step in order to obtain very reasonable results.

Note that in the above splitting algorithm only equations (4.21), (4.22), (4.23), and (4.25) constitute boundary value problems for PDE's and in order to discretize them with finite elements we need to derive proper weak formulations. As it is usual for finite element projection schemes [38] we choose $\mathbf{u}_1^*$ to be in $\mathbb{H}_0^1(\Omega)$ and p_1 to be in $H^1(\Omega)$. The final velocity of **Substep 3**, $\mathbf{u}_1^{**}$, is also projected onto $\mathbb{H}_0^1(\Omega)$ as discussed in [38]. The essentially new element in the present scheme (compared to the one suggested by Glowinski et al. [33]) is that the Lagrange multiplier $\boldsymbol{\lambda}$ is no longer distributed but rather a globally (over the entire Ω) defined quantity. It is also a solution to a boundary value problem similar to the problem for $\mathbf{u}_1$ and therefore it is naturally chosen to be in $\mathbb{H}_0^1(\Omega)$. Actually it plays the role of a correction to both, the fluid and particle velocities. Of course, this correction will increase the divergence of the fluid field and in order to better control the divergence of this correction one may decompose the interaction force $\mathbf{F}$ into a divergence free part, $\boldsymbol{\lambda}$ and a curl free part, ∇p_2, and define both to be a solution of a Stokes-like problem. This would require to do one projection step for the additional pressure p_2 at each iteration for imposition of the rigid body motion. The present algorithm seems to be reasonably stable as it is, and therefore we did not try this more expensive alternative.

The weak formulations of (4.21) and (4.22) are given by

Find $\mathbf{u}_1^* \in \mathbb{H}^1(\Omega)$ *such that*

$$\int_\Omega (\rho_1\tau_0\mathbf{u}_1^*\mathbf{v} + \mu_1\nabla\mathbf{u}_1^*\nabla\mathbf{v})d\Omega = -\int_\Omega \rho_1(\tau_1\tilde{\mathbf{u}}_1^n + \tau_2\tilde{\mathbf{u}}_1^{n-1})\mathbf{v}d\Omega + \int_\Omega p_1^n\nabla\cdot\mathbf{v}d\Omega, \quad \forall\mathbf{v} \in \mathbb{H}_0^1(\Omega). \tag{4.27}$$

Find $p_1^{n+1} \in H^1(\Omega)$ *such that*

$$\int_\Omega \nabla(p_1^{n+1} - p_1^n)\nabla q d\Omega = \tau_0 \int_\Omega \rho_1\mathbf{u}_1^*\nabla q d\Omega \quad \forall q \in H^1(\Omega). \tag{4.28}$$

Find $\mathbf{u}_1^{**} \in \mathbb{H}^1(\Omega)$ *such that*

$$\int_\Omega \rho_1 \tau_0 (\mathbf{u}_1^{**} - \mathbf{u}_1^{*}) \mathbf{v} d\Omega = \int_\Omega p_1^{n+1} \nabla \cdot \mathbf{v} d\Omega, \quad \forall \mathbf{v} \in \mathbb{H}_0^1(\Omega). \tag{4.29}$$

As discussed by [38] the last step can be skipped but it usually reduces the divergence of the so computed velocity field. In the present context, since we do not control the divergence of the iterative correction $\boldsymbol{\lambda}$, it is advisable to solve (4.29). Before we give the weak formulation of (4.25) we note that if we choose $\alpha = \rho_1 \tau_0$ the problem (4.23) can easily be resolved and the solution is

$$\mathbf{u}_1^{k+1,n+1} = \mathbf{u}_1^{k,n+1} + (\boldsymbol{\lambda}^{k+1,n+1} - \boldsymbol{\lambda}^{k,n+1}).$$

Thus, this choice for α is very natural. Moreover, this choice yields an inner product for imposition of the rigid body motion which is a compromise between the $\mathbb{H}^1$ and the $\mathbb{L}^2$ inner products, and given by $\rho_1 \tau_0 (.,.)_0 + \mu_1 (\nabla ., \nabla .)_0$ where $(.,.)_0$ denotes the usual $\mathbb{L}^2$ inner product.

Finally, the weak formulation of (4.25) is given by

Find $\boldsymbol{\lambda}^{k+2,n+1} \in \mathbb{H}^1(\Omega)$ *such that*

$$\begin{aligned}
&\int_\Omega (1 + \frac{\rho_1}{\rho_{2,i} - \rho_1})(\rho_1 \tau_0 \delta_t \boldsymbol{\lambda}^{k+2,n+1} \mathbf{v} + \mu_1 \nabla \delta_t \boldsymbol{\lambda}^{k+2,n+1} \nabla \mathbf{v}) d\Omega \\
&= \sum_{i=1}^{n} \int_{\Omega_{2,i}} \rho_1 \tau_0 (\delta_t \mathbf{u}_1^{k+1,n+1} - \delta_t \mathbf{u}_2^{k+1,n+1}) \mathbf{v} \\
&+ \sum_{i=1}^{n} \int_{\Omega_{2,i}} \mu_1 \nabla (\delta_t \mathbf{u}_1^{k+1,n+1} - \delta_t \mathbf{u}_2^{k+1,n+1}) \nabla \mathbf{v} d\Omega \\
&- \mu_1 \sum_{i=1}^{n} \int_{\partial\Omega_{2,i}} \mathbf{n} \nabla (\delta_t \mathbf{u}_1^{k+1,n+1} - \delta_t \mathbf{u}_2^{k+1,n+1}) \mathbf{v} ds \quad \forall \mathbf{v} \in \mathbb{H}_0^1(\Omega).
\end{aligned} \tag{4.30}$$

The surface integral in the right hand side is a result of the integration by parts of the Laplacian over each of the particles. The numerical experience shows that this integral as well as the surface integral in the first equation of (4.24) are usually very small and do not seem to alter significantly the results in case of spherical particles. Therefore, they are not taken into account in the results of section 3.

The formulations (4.27)-(4.30) are discretized by means of $\mathbb{P}_2 - P_1$ finite elements using $\mathbb{P}_2$ interpolation for the velocity and $\boldsymbol{\lambda}$ and P_1 interpolation for the pressure. The grid is a fixed tetrahedral Eulerian grid that discretizes Ω. The resulting linear systems are solved by means of a parallel version of the conjugate gradients method.

To complete the discussion on the numerical algorithm, we need to specify the quadrature rule that is used to compute the integrals in these formulations. These are, of course, a Gauss type quadratures that ensure the exactness of the integration. Only the integrals in the right hand side of (4.30) can cause some problems since the domain of integration is not (in general) exactly covered with finite elements. In case that the surface of the particle intersects the interior of a given element we adjusted the Gaussian integration weights of the Gauss points inside the particle so that their sum is equal to one. This is a simple procedure that is not exact anymore but makes the corresponding quadrature consistent. More sophisticated solutions like adaptive integration procedures [15] can also be used but this simple fix seems to work well on reasonably fine grids.

4.3 Conclusions

The method that we suggested in the present paper is a fictitious domain Eulerian method for direct simulation of particulate flows. The main advantages in comparison to other methods of the same type (in particular the method that we used as a starting point for the present study, proposed by Glowinski et al., [33]) are that:

1. It defines a global Lagrange multiplier, $\boldsymbol{\lambda}$, whose physical meaning is of an additional velocity field that imposes the rigid body motion. This allows us to completely eliminate the Lagrangian grids used by [33] to discretize the distributed Lagrange multipliers in their case. This allows for the use of basically only one solver on relatively regular grids which facilitates the parallel implementation of the method.

2. The global Lagrange multiplier, together with a proper choice for the inner product that is used to impose the rigid body constraint allow to avoid the need to solve the linear system for computing the distributed Lagrange multipliers in each particle. Although this system is not large, it will eventually be solved on each iteration of **Substep 4** of the algorithm above. If the flow contains many particles, the savings can be significant.

3. The present algorithm employs a second order in time, incremental projection scheme for the resolution of the generalized Stokes problem which, as our numerical experience shows, performs significantly better than a first order scheme

for almost the same expenses. The spatial discretization is also second order accurate for the velocity and first order accurate for the pressure.

Chapter 5

Computational implementation

5.1 Collision strategy

For handling more than one particle, a collision treatment mechanism had to be added to prevent particles from interpenetrating each other. To avoid this overlapping a repulsive force was added to equation (4.10). Two different repulsive forces were tried. The first one was of the form suggested by Glowinski et al. [33], where a short-range repulsion force between particles that are near contact is introduced. For two given particles i and j the force is calculated as:

$$F_R = \frac{c_{i,j}}{\epsilon} \frac{(d_{i,j} - R_i - R_j - \rho)}{\rho} \frac{G_i G_j}{d_{i,j}} \tag{5.1}$$

Where $c_{i,j}$ is a scaling factor, ϵ is a small positive number, $d_{i,j}$ is the distance between the center of the two particles, R_i, R_j are the radius of the particles, ρ is the range of the repulsion force and G_i, G_j are the center of mass of both particles.

In this approach the choice of the scaling and stiffness parameters ($c_{i,j}$ and ϵ) is very important and in general the ideal values of these parameters may vary for different cases. Since the factors $c_{i,j}$ and ϵ had to be changed for every case, a more general approach was considered.

In this new approach the repulsive force is calculated in such a way that it would compensate the velocity so that the particles do not overlap. The collision algorithm first checks if the particles' separation is larger than a given threshold related to the particle's radius and the mesh resolution. If the distance is less than this value then the repulsive force is calculated iteratively so that both particles move along the line that passes through the center of mass of both particles and that the minimum

distance is still maintained.

The algorithm can be summarized as follows (see Figure 5.1):

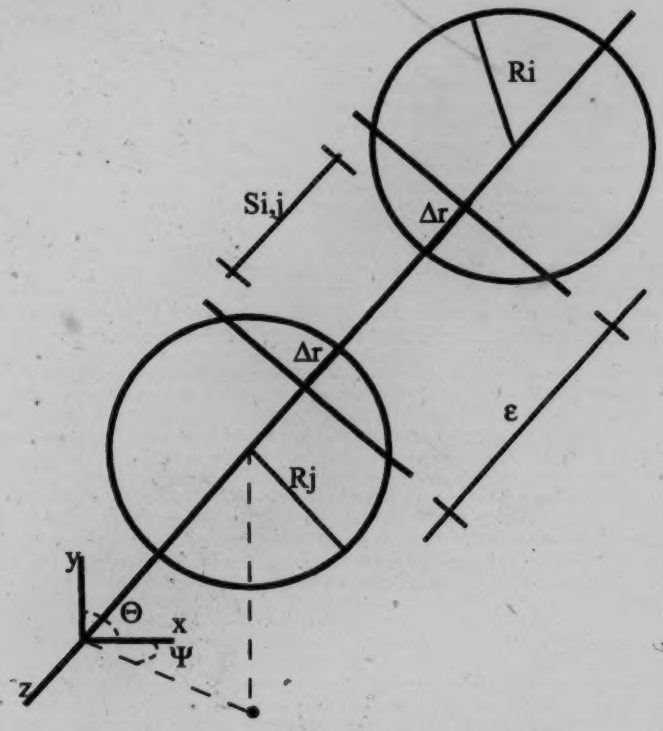

Figure 5.1: Two particles near collision.

1- Estimate the particle's position $\mathbf{X_i}$ with the calculated velocity as in equation (4.26).

2- Calculate separation between particles and detect possible collision.

$$s_{i,j} = |\mathbf{X_i} - \mathbf{Xj}| - (R_i + R_j) \tag{5.2}$$

Where $s_{i,j}$ is the separation between two spherical particles i, j, $\mathbf{X_i}$ and $\mathbf{Xj}$ are the center of mass position of the i and j particle and R_i and R_j are their radii. If $s_{i,j}$ is less than a minimum separation ϵ then the each particle is moved a distance $\Delta\mathbf{r}$ necessary to maintain that security distance.

3- Establish the distance needed to maintain the minimum admissible separation along the direction of the line that passes through both center of mass. This direction

is given by θ and ψ.

$$\Delta \mathbf{r}_{i,(\theta,\psi)} = \frac{M_i(\epsilon - s_{i,j})}{M_i + M_j} \tag{5.3}$$

Where M_i and M_j are the masses of particle i and j respectively.

4- The velocity needed to move the particle i a distance $\Delta \mathbf{r}_{i,(\theta,\psi)}$ $\mathbf{U}_{i,(\theta,\psi)}$ is then

$$\mathbf{U}_{i,(\theta,\psi)} = \frac{\Delta \mathbf{r}_{i,(\theta,\psi)}}{\delta t} \tag{5.4}$$

5- Update the particle's velocity adding this velocity to the current one.

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^{i+1} + \mathbf{U}_{i,(\theta,\psi)} \tag{5.5}$$

6- Update the particle's position.

In order to impose that the fluid velocity in the region occupied by the particles is equal to the new particle's velocity we need to perform **Substep 4** of the splitting algorithm above. However, if we choose ϵ to be very small, of the order of δt^2, then the fluid velocity would be perturbed with a term which is of the order of the time discretization error and we can skip this additional iteration.

This algorithm is efficient in preventing collisions. It basically maintains a security layer to avoid particle overlapping. Physically speaking, it is well known that before the two particles touch each other, the thin film formed between them should drain completely. This is a physical phenomenon of a (time and spatial) scale different from the scale of the macro-motion described by the equations given above. Therefore, a more careful approach would require the development of a separate model for the film drainage which has a very different time scale and then merging it with the macro-model considered in the present paper. The present collision prevention approach presumes that the thickness of the film between any two particles is of the order of δt^2.

5.2 Element Searching Algorithm

In the implementation of the method of characteristics an algorithm to locate the element that contained the foot of the characteristic had to be used. The element searching algorithm used was that proposed by Minev and Ethier [54].

When solving the convection part of the Navier-Stokes equations the foot of the characteristic has to be found. When the time steps is small, so that the local Courant number is small (of order one) then the foot of the characteristic is close to its terminal

point, x. In this case, a good guess for the element containing the foot is available. On the other hand, at large time steps, the foot of the characteristic can be located relatively far from its terminal point, and a good guess is not easily available. Such a guess is crucial for the effectiveness of the whole algorithm. The algorithm to estimate such a guess and conduct the subsequent search is described here.

Let's consider a computational domain Ω, that has been discretized using an unstructured grid

$$G = \{P_m; P_m \in \Omega\}_{m=1}^{N_g} \tag{5.6}$$

where G is the unstructured, P_m is a point belonging to that mesh and N_g is the total number of points of this mesh. For such a mesh, the local Courant number is defined as:

$$\mathrm{Cu} = U_{P_i} \frac{\Delta t}{\Delta x} \tag{5.7}$$

with U_{P_i} the velocity at point P_i and Δx the minimum inter-nodal spacing among the elements containing P_i. The Courant number measures the distance traveled by advection during one time step. G is overlayed with a structured grid

$$C = \{V_n\}_{n=1}^{M} \tag{5.8}$$

consisting of uniform cubes (voxels, V_n) with a side length equal to the maximum possible spacing d between two neighboring nodes in G. This structured grid C is a parallelepiped. The idea for such an auxiliary grid was originally sketched out by Buscaglia and Dari [11]. This auxiliary grid may contain a huge number of cells. To avoid storing empty cells that could mean using large amount of memory and reducing the overall performance of the algorithm all information is stored in compact storage format in which all the empty voxels of C are excluded, and only non-empty voxels are considered. This requires two related data structures that allow rapid searching for the nearest grid point P_m to a given point within the computational domain, P^*. To describe the necessary data structures we introduce the following notation:

- Without loss of generality we assume that the x-axis is parallel to the longest side of C. We denote the number of voxels in the x, y and z directions by N_x, N_y and N_z respectively, and the total number of non empty voxels of C by N_g. A region of C is said to be empty if it contains no nodal points P_m.

- Each voxel is identified with three integers accounting for its position in C in the x, y and z directions. If x_0, y_0, z_0 are the physical coordinates of the origin

of C, then the voxel coordinates (i^*, j^*, k^*) of the point P^* are given by the integer parts of:

$$\begin{cases} i^* = \frac{(x^* - x0)}{d} \\ j^* = \frac{(y^* - yo)}{d} \\ k^* = \frac{(z^* - z0)}{d} \end{cases} \tag{5.9}$$

Each voxel is then associated with a single label given by

$$label(i, j, k) = iN_yN_z + jN_z + k \tag{5.10}$$

The data structures, created and filled in a preprocessing step, designed to map voxel coordinates (i, j, k) to a list of unstructured grid points P_m contained in that voxel are:

- A vector **B** of dimension N_g containing an ordered list of the node numbers of the grid G.
- A two-dimensional matrix **A** that has N_x rows and is of variable size in the second direction. **A** contains N_x x-slices of the structured mesh C each of which is composed of "y-columns" of voxels. The number of entries in row i of **A** is equal to the number of non-empty y-columns in the ith x-slice of C. The entries of **A** contains a pair *(k-pair, offset)*, where *offset* is an index to the vector **B** of the grid point P_m in voxel *(i, j-index, k-index)*.

The searching algorithm works as follows:

1. Given a point $P^*(x_1, x_2, x_3)$, determine its voxel coordinates (i^*, j^*, k^*) in the structured grid C.
2. Perform a search in the i^* row of **A** to determine which entry has *j-index* equal to j^*.
3. Perform a search for the corresponding *k-index* in the vector of the entry identified in step 2.
4. Using offset to reference the appropriate node numbers in **B**, find the closest point P_m to P^* among the points in the voxel to which P^* belongs. Then P^* belongs to one (or more) of the elements containing P_m. These elements are searched to determine which of them contains P^*.

The computational expense for this algorithm is of order $log(N_y N_z)$ per point P^*. Then, the total algorithm cost scales as $N_g log(N_y N_z)$, which varies nearly linearly with the number of nodes in the mesh. A naive search for the closest point performed by measuring the distance from each point to P^* requires order $N_g(N_g+1)/2$ operations.

Remarks:

- The above algorithm is useful only if the local Courant number is larger than 1. If Cu is smaller than 1, the foot of the characteristic will be contained within one of the elements containing its terminal point, and the terminal point is then a good initial guess. The efficiency of the above algorithm integration is virtually independent of Cu.

- If the foot of the characteristic is not found in any of the elements that contain the initial guess (computed as described above), then the time step is subdivided into substeps Δs and the above algorithm is applied within each substep. If $\mathbf{X}_{\mathbf{x},l}^{n+1}$ is the foot of the characteristic at the end of the l-th substep, the velocity $\mathbf{u}(\mathbf{X}_{\mathbf{x},l}^{n+1})$ needs to be updated by extrapolating the velocity field at time level $(n+1)\Delta t - l\Delta s$. Choosing Δs small ensures that the foot of the characteristic $\mathbf{X}_{\mathbf{x},m}^{n+1}$ at time $m\Delta s = \Delta t$ is always within the computational domain Ω.

5.3 Conclusions

In this chapter we addressed necessary issues for the correct implementation of the code. A fast searching algorithm is required for the method of characteristics as an integration method to solve for the convection part of the Navier-Stokes equation. At each time step the feet of the characteristics had to be found and the described algorithm was used for that purpose.

For handling multiple particles a collision strategy was developed. It maintains a security zone between particles so that they do not overlap.

Chapter 6

Parallel implementation and other computational considerations

6.1 Parallel linear solver

The direct numerical simulation of particulate flow is a computationally intensive problem, since we are finding the solution of non-stationary equations to describe a flow that evolves in time and over many timesteps. To find the solution at each timestep a nonlinear system of equations must be solved, and within each nonlinear iteration a linear system is solved. Most of the computing resources needed by this kind of finite element codes are spent in the linear solvers.

The time-stepping is handled by a splitting operator technique as described before. Each substep requires the solution of a linear system of the form $\mathbf{Ax} = \mathbf{b}$. The linear systems are large, sparse, symmetric, and definite. For this reason special structures have to be designed to store the matrices of this system. For large linear systems using a direct method is not a practical solution since the memory requirements become unreasonable. Iterative solvers are a natural solution. In our simulations up to 91% of the CPU time of the entire simulation is consumed by the linear system solver alone. For this reason we focus on addressing this problem, trying to implement a parallel solver for a shared memory computer. All the parallel code was done in OpenMP and optimized for an SGI Origin model 2400, with 64 processors (400MHZ), 16384 Mbytes of main memory size, 32 Kbytes of data cache size. For the current algorithm, the matrices are symmetric and positive definite so we choose the linear conjugate gradient (CG) as our linear solver.

```
1  x_0 = initial guess, r_0 = b - Ax_0, d_0 = r_0, δ_0 = r_0^T P r_0
2  do i= 1..Nmax
3     h_i = A d_{i-1}
4     γ_i = d_{i-1}^T h_i
5     τ_i = δ_{i-1}/γ_i
6     x_i = x_{i-1} + τ_i d_{i-1}
7     r_i = r_{i-1} + τ_i h_i
8     δ_i = r_i^T P r_i
9     check for convergence
10    if ( not converged )
11       β_i = δ_i/δ_{i-1}
12       d_i = β_i P d_{i-1} + P r_i
13    else
14       quit
15 end do
```

Figure 6.1: Conjugate gradient algorithm for solving $\mathbf{Ax} = \mathbf{b}$.

The CG method minimizes the A-norm of the error over the current subspace. Figure 6.1 shows the pseudo-code for the CG algorithm. The method proceeds by successively approximating the solution, computing the corresponding residuals to the current solution and computing directions used to update the iterates and the residuals. It can be shown that an upper bound for the number of iterations steps required to get the A-norm of the error below some prescribed value ϵ is proportional to the square root of the condition number of A and the speed of convergence depends on the spectral properties (such as eigenvalues distribution) of this matrix [25]. In an attempt to improve these properties the CG is used in combination with some kind of preconditioning. The idea is to construct a matrix $\mathbf{K}$, such that $\mathbf{Kx} = \mathbf{b}$ is much easier to solve than $\mathbf{Ax} = \mathbf{b}$. Normally, $\mathbf{K}$ is constructed to be an approximation of $\mathbf{A}$. A first attempt was to obtain $\mathbf{K}$ by multiplying the $\mathbf{A}$ matrix with a matrix $\mathbf{P}$ obtained from the inverse diagonal of $\mathbf{A}$.

The conjugate gradient method involves one matrix-vector product, three vector updates ($\mathbf{x}$,$\mathbf{r}$,$\mathbf{d}$), and two inner products (τ and β) per iteration.

In the above algorithm there are two obvious synchronization points, namely the

computation of the two scalar constants τ and β. The algorithm can only update $\mathbf{x}_i$ and $\mathbf{r}_i$ after completing the inner product for τ_i. The same observation applies to updating $\mathbf{d}_i$.

To calculate the global vector dot product all the local contributions need to be summed. This is done using an OpenMP atomic update. Communication is also required after the matrix-vector product.

Steps 6, 7 and 12 do no require communication. Steps 6, 7, and 8 can be combined: the computation of a segment of 6 be followed immediately by the computation of a segment of step 7, and this can be followed by the computation of a part of the inner product in 8. This saves on load operations for segments.

6.2 Other computational considerations

Vector updates were trivially parallelized by dividing and assigning segments of the vector to each processor. Matrix-vector products were parallelized by splitting the matrix in groups of rows. Since the matrices are extremely large and sparse compact storage techniques were implemented. For the serial version of the code the nodes were rearrange considering boundary conditions. In this way, all the rows corresponding to prescribed nodes could be eliminated from the global matrices and the systems could be reduced.

6.3 Conclusions

Most of the CPU time used by the numerical solver is spent in the linear solver. Since the systems are large and sparse the conjugate gradient method was implemented as the linear solver. This was successfully parallelized. The parallel routines make no assumption about the structure of the grid. Further improvements to increase the parallel performance of the code could involve implementing a domain decomposition on structured grids.

Chapter 7

Experimental velocity measurement of settling spheres

7.1 Experimental setup

To validate the numerical code experiments were conducted to measure the velocity of settling spheres under different conditions. A tank of 40cm × 68cm × 33cm was built as shown in Figure 7.1.

Glycerin and Drakeol 7 mineral oil were used for the different experiments. Table 7.1 shows the physical properties for these fluids. Steel spheres of 0.99, 1.99, 2.20 and 2.38 mm of diameter and 4.75 mm nylon diameter balls were used. The densities were: $\rho_{steel} = 7800$ kg/m^3 and $\rho_{nylon} = 1140$ kg/m^3 respectively.

The particles were released and the their motion was captured with a Redlake high speed digital camera that captured frames at intervals of 0.004 seconds. A strong light was used to illuminate the particles. Since the particle terminal velocity determined the shutter rate of the camera, the particles were chosen so that their terminal velocity would allow a rate that would produce images with little blur and the particles would be still well lit. The other variable affecting the image brightness

Fluid	Density 26C (kg/m^3)	Viscosity (cPoise)
Glycerin	1177	11.7
Drakeol	826	17.75

Table 7.1: Physical properties of the fluids

was the aperture. This had to be adjusted according to the focus range needed and the amount of light necessary to make the pictures bright enough.

The motion of the particle was then studied by analyzing the different frames of the acquired data. To analyze the images Redlake Motionscope Media Player was used. The imaging software allowed us to measured distances and velocities. Figure 7.2 shows and example of an image captured with the high speed camera.

While conducting these experiments two main problems had to be overcome. The first one was the particle releasing mechanism. A method had to be devised in order to be able to release the particle without disturbing the fluid. This method had to be capable of handling more than one particle. The second problem encountered was that the images provided 2-D information only. For some cases this was insufficient to reveal the motion of the particles. An example of this was the case of two particles drafting, kissing and tumbling. For images showing the particles settling together there was the uncertainty if this was so (they were really touching each other) or if they were at the same vertical position but at different depths. For validating particle-particle interaction this information was very important. The solutions implemented to overcome these problems are discussed in the next sections.

7.2 Particle releasing mechanisms

To minimize the disturbance produced while releasing the spheres three different methods were tried: tweezers, a vacuum system, and an electromagnet.

1. **Tweezers**
   At the beginning particles were released using a pair of tweezers. The tweezers were good enough for simple, one particle drops at high Reynolds numbers ($Re > 500$). For these velocities, the error caused by the particle clinging and rolling off the tweezers arms was slight and experiments were highly repeatable. At low Re, however, error was substantial and velocity profiles would often differ by as much as 20%.

2. **Vacuum system**
   Our next attempt was to develop a vacuum system that could remotely release two balls at varied distances by the use of a suction bulb on a long vacuum line. The problem with the system was that the two suction holes were very difficult to machine to a perfect fit on the spheres. The fluid was often sucked past the

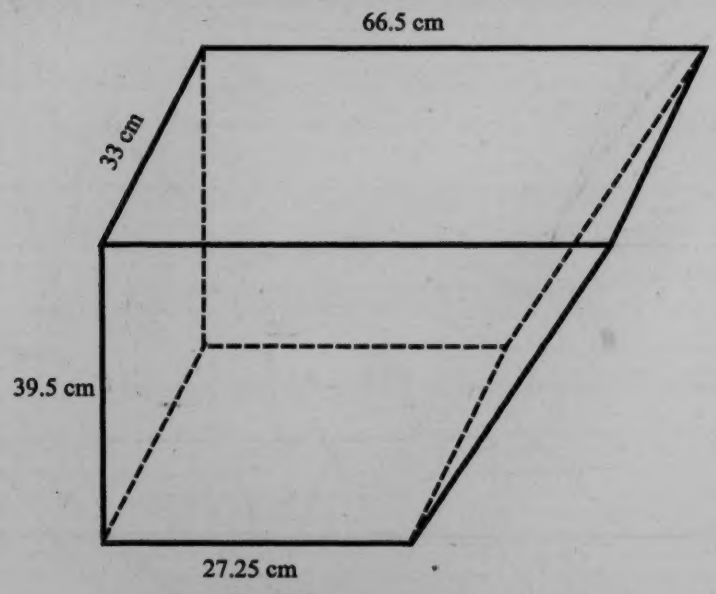

Figure 7.1: Experimental setup.

Figure 7.2: Analysis of data captured with a high speed camera.

spheres and into the vacuum lines, which tended to rush out upon release causing substantial error in the experiment. This vacuum system worked well only for dropping very light nylon and plastic balls into a glycerin/water mixture.

3. **Electromagnet**
   Finally, an electromagnet was designed. Unfortunately, this meant that only steel ball bearings would have to be used. For this kind of particle a more viscous fluid was required. Mineral oil Drakeol 7, was used for the experiments with the steel bearings. An Al/Ni/Co electromagnet was assembled in a c-shaped iron core with an adjoining permanent magnet. A short burst of current was used to change the polarity, which changed the flux path and gave a near residual-free release of the particles. We found the magnet to work best with an energizing current of about 2.0 amps and a release current of about 1.0 amp. Since the coil impendence was 20.9 ohms, a power supply capable of about 45 volts was used.

7.3 2D vs. 3D measurements

Two dimension videos often did not give an adequate representation of the particles' movement, so a three dimensional view was devised using mirrors. The 3D information was particularly important when analyzing particle-particle interactions Two mirror were placed inside the tank. A calibration plate was built so that the mirrors could be places in the correct position at the beginning of each experiment. Figure 7.3 shows an example of an image obtained using this technique.

The images obtained in this way were analyzed in a similar way as before but now 3-D coordinates of the particles could be obtained. The vertical displacement could also be matched for the different views. Some of the 2-D and 3-D data obtained experimentally is presented in Chapter 8 and compared with the results obtained with the numerical simulations.

7.4 Conclusion

In order to validate the numerical model, experiments were conducted to measure the settling velocity of solid particles settling as well as to obtain data describing particle-particle interaction. For this purpose a tank was constructed and fluid and

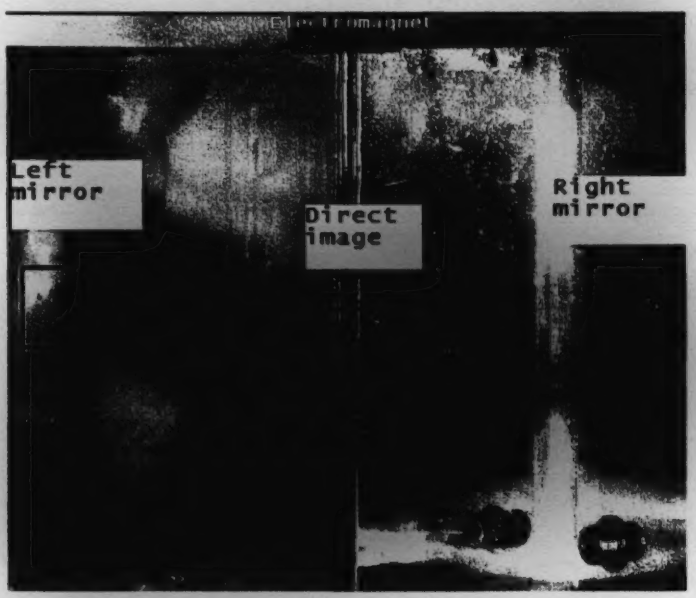

Figure 7.3: Experimental setup: 3D information retrieval.

particles were selected in order to match the capability of the available hardware. Data was acquired using a high speed camera. During the experimental work two major problems had to be overcome:

- Devising a suitable particle releasing mechanism that would not introduce disturbance in the fluid and would be capable of handling multiple particles.
- Obtaining 3-D data.

Both problems were successfully solved and the necessary experimental data was obtained to validate our numerical simulations.

Chapter 8

Numerical and experimental results: single particle modeling

The program implemented using the fictitious domain method with global Lagrange multipliers was tested for different cases. The first trials were those involving the sedimentation of a single particle under gravity in a fluid originally at rest. The different program parameters were varied (i.e. Reynolds and Froude number, timestep, density ratio, mesh resolution) and the results validated with experimental data. Other test performed involving only one particle included influence of the distance to the wall of the channel on terminal velocity and wall collisions. At a later stage, the study focused on situations where more than one particle are present. The simplest cases involving more than one solid particle are those where particles do not interact with each other. Particle-particle interaction was taken into account and the program was used to simulate collisions. In this chapter we present the results of some of the simulations starting with those involving one particle. Simulations involving more than one particle are presented in next chapter. All the numerical simulations are presented using dimensionless variables.

8.1 One particle settling in a vertical channel

The first case to be tried was that of a single particle settling under gravity in a fluid initially at rest. The program was tested for different Reynolds numbers and density ratio. In most cases the difference between the experimental terminal velocity and the one calculated using the program did not differ in more that 7%. Figure 8.1 shows the

the experimental values obtained for a steel sphere of radius 1.99 mm and density $\rho_{\text{Solid}} = 7800$ kg/m^3 settling in a fluid with density $\rho_{\text{Fluid}} = 826$ kg/m^3 and viscosity $\mu = 17.8$ cPoise. The terminal velocity for the sphere in this case was 30.5 cm/s. Based on this terminal velocity the calculated value of the Reynolds number was Re=28.7 and the Froude number was Fr=5.0. The experiment was repeated 3 times and the error between trials was around 4%. To compare with the numerical results,

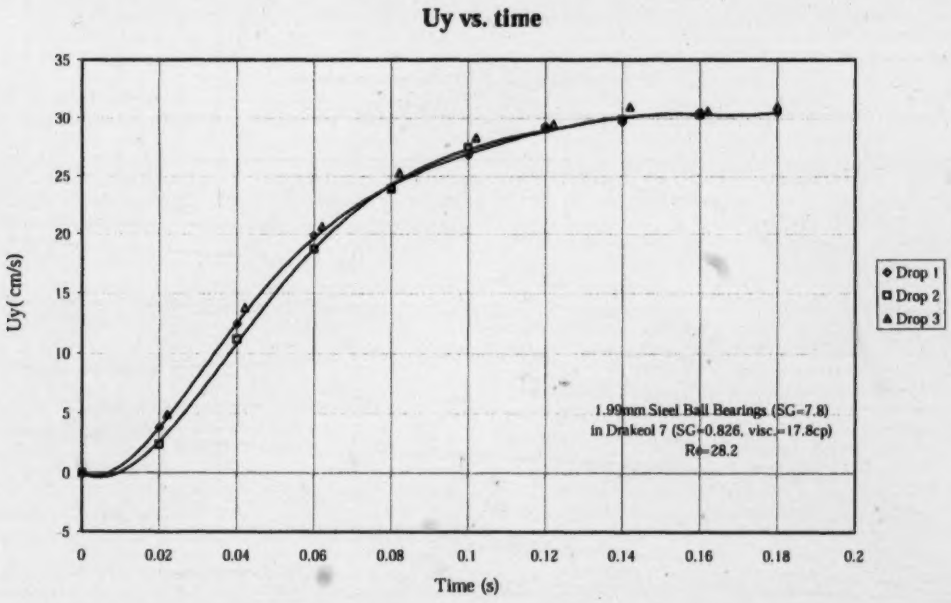

Figure 8.1: One particle settling. Experimental vertical velocity vs. time. Re=28.7.

the experimental results were made dimensionless. The experimental velocity was divided by the terminal velocity, U_{inf}, and the time by the ratio of the particle's diameter, D_{sp}, and its terminal velocity. The particle's displacement X was made dimensionless by dividing by the particle's diameter:

$$U_{dimensionless} = \frac{U_{experimental}}{U_{\text{inf}}} \tag{8.1}$$

$$X_{dimensionless} = \frac{X_{experimental}}{D_{sp}} \tag{8.2}$$

$$t_{dimensionless} = \frac{t_{experimental}}{D_{sp}/U_{\text{inf}}} \tag{8.3}$$

The procedure to calculate the dimensionless variables was repeated throughout this thesis whenever a direct comparison between experimental and numerical results was needed.

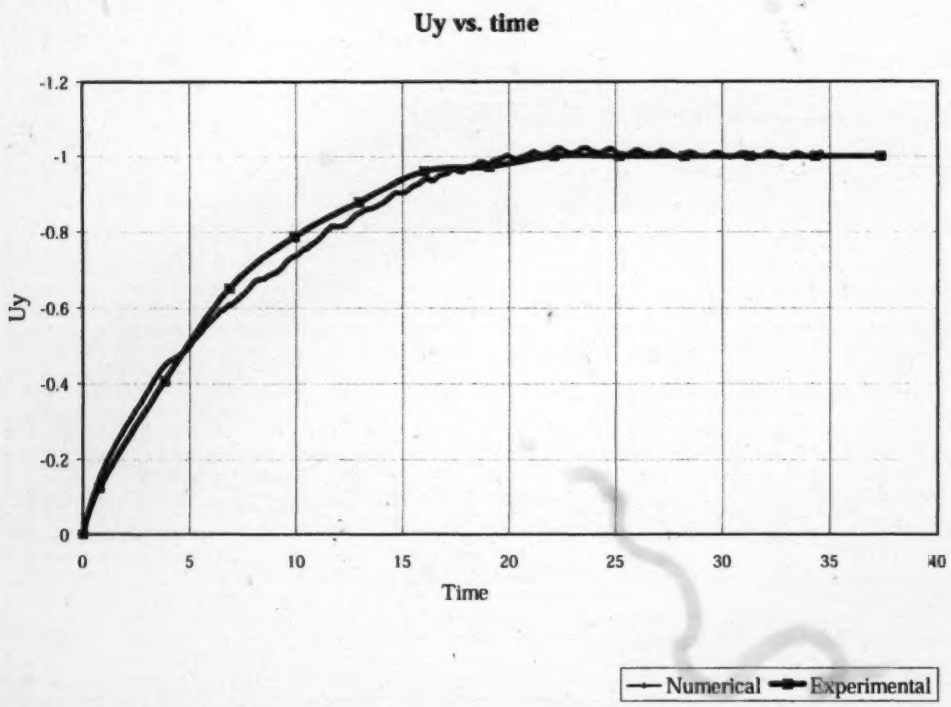

Figure 8.2: One particle settling. Comparison of experimental and numerical vertical velocity vs. time (dimensionless). Re=28.7.

Figure 8.2 shows the comparison of the experimental velocity and the results for a simulation resembling previous experiment. For the numerical simulation the Reynolds number was set to 28.7 and the Fr number was 5.0. The mesh used for this case was a rectangular channel with a cross sectional area of (4×4) and a length of 12. The mesh had 306231 nodes and 205800 elements. The solid particle had a diameter of 1. The ratio of densities was set to 9.44. The timestep used was of 0.01.

The particle accelerates immediately after it is released but eventually reaches an approximately constant sedimentation velocity when gravity balances buoyancy and

the drag force. At constant terminal velocity some oscillations occurred. Figure 8.2 shows that the numerical results are in agreement with the experimental data.

A similar experiment was conducted for a steel bearing of diameter 0.99mm. The terminal velocity in this case was 13.2 cm/s. The Reynolds number was Re=6.08 and the Froude number Fr=1.8. Figure 8.3 shows the experimental results (vertical velocity versus time).

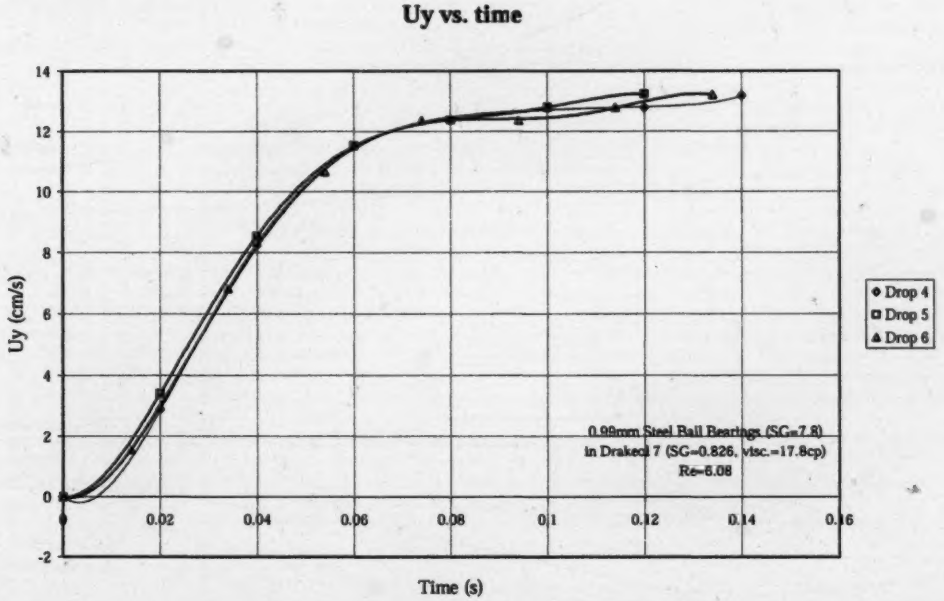

Figure 8.3: One particle settling. Experimental vertical velocity vs. time. Re=6.08.

The experimental results were used to carry out a simulation using the calculated dimensionless numbers. For the simulations the mesh was the same as in the previous case (306231 nodes and 205800 elements). The timestep was set to 0.01. The fluid density was $\rho_{\text{Fluid}} = 826$ and the particle's density $\rho_{\text{Solid}} = 7800$. The sphere's radius was 0.5. Figure 8.4 shows the evolution of the vertical velocity with time for this simulation and for the experimental set up. All variables were made dimensionless following the procedure explained for the previous case.

The numerical results agree with the experimental ones.

Our simulation results were also compared to that published by Mordant et al. [57]. Figure 8.5 shows the results obtained for a sphere of radius = 0.5 settling in a

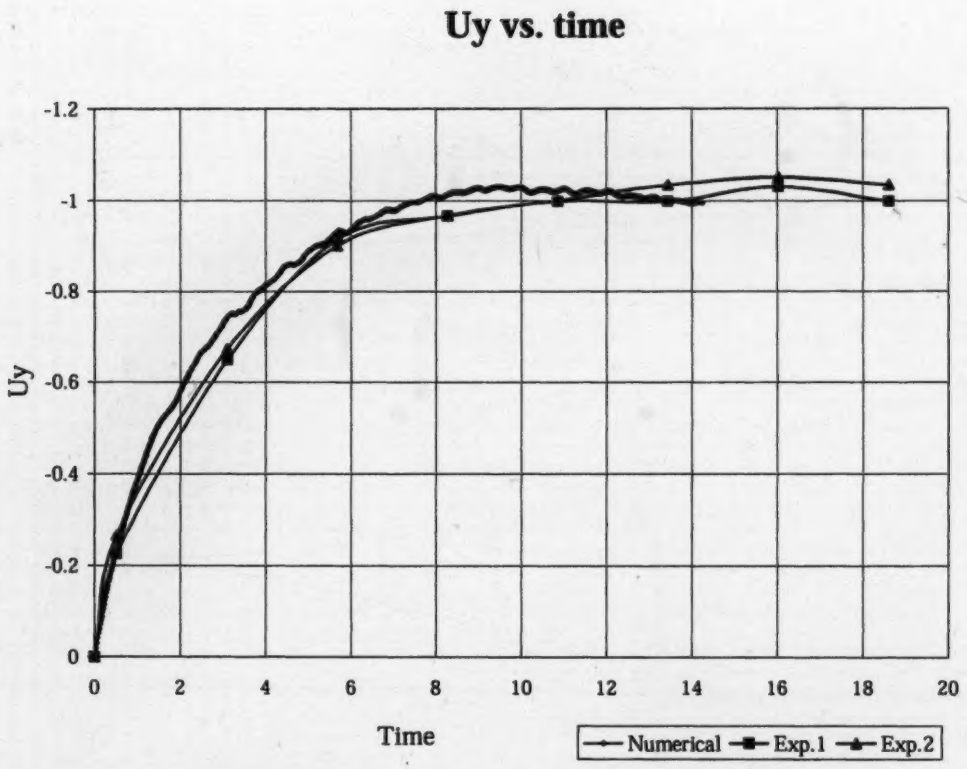

Figure 8.4: One particle settling. Experimental and numerical vertical velocity vs. time (dimensionless). Re=6.0.

rectangular channel with Re=41 and Fr=1.12. The channel's mesh is the same as the one described above. The time step is set to 0.01.

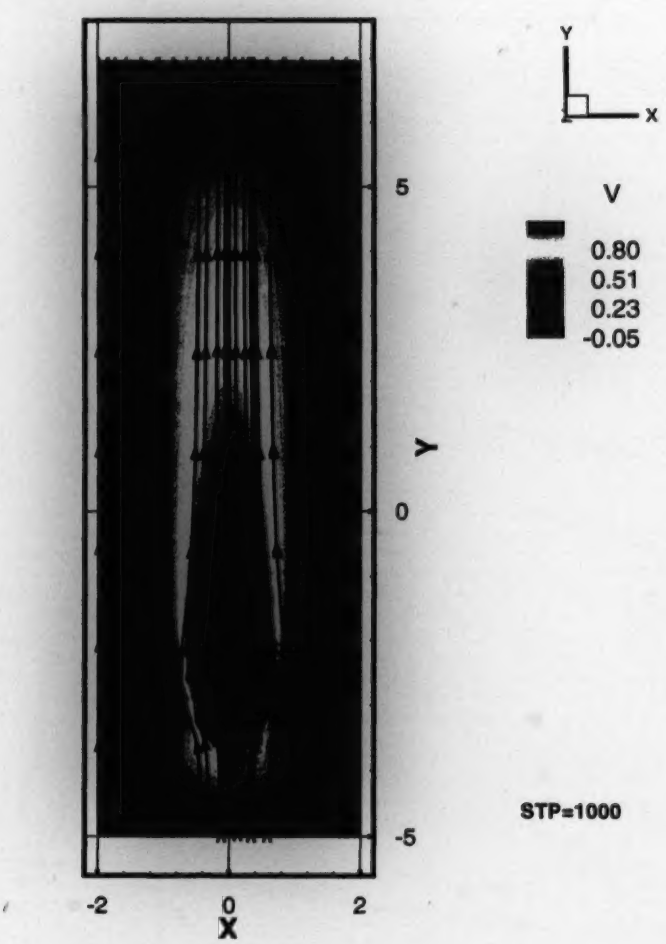

Figure 8.5: One particle falling under gravity. Re=41.

In figure 8.6 the velocity obtained with the simulation is compared to that presented by Diaz-Goano et al. [23] and to the curve proposed by Mordant et al. [57]. The latter authors experimentally studied the motion of a solid sphere settling under gravity in a fluid at rest. All their curves collapsed onto a single exponential shape

given by:

$$\frac{v_{exp}}{V_{Terminal}} = 1 - e^{(-\frac{3t}{\tau_{95}})}$$

where v_{exp} is the vertical velocity of the particle, $V_{Terminal}$ is the terminal velocity, t is the time, and τ_{95} is the time it takes the particle to reach 95% of the limit speed. The shape of our simulation curve matches closely to the one produced using the equation presented by Mordant [57] and to the one in [23]. For this comparison the code presented in [23] used an auxiliary mesh for the sphere with 76 nodes and 226 elements. The input file specifying geometry and connectivity tables and other parameters such as Reynolds number, Froude number, time step, particle's radius was the same for running both codes: 179401 nodes and 120000 elements, dt=0.01, Re=41, Fr=1.12. The time required to complete 100 time steps of this simulation was 20% less with the new algorithm. This time difference is expected to be even larger when simulating more than one particle. The speedup is a direct consequence of defining the Lagrange multipliers globally and expanding them over the same basis functions as the fluid and solid particle velocity fields.

8.2 One particle settling at higher Reynolds number

Figures 8.7 and 8.8 show the streamlines for the case of a sphere settling with a higher Reynolds number (Re=118.0 and Fr=0.01). At this Reynolds number a pair of vortices can be clearly seen behind the sphere. From our numerical simulation we observed that the diameter of the vortices formed increased with increasing Reynolds numbers. For a three dimensional results it was very difficult to measure exactly the length of this vortices. In this case, the recirculation wake is close to one diameter long. This can be compared qualitatively with results presented by Taneda [68]. Taneda's experimental work shows the development of the attached wake behind a rigid sphere. He presents a plot of dimensions of the attached wake as a function of the Reynolds number. From that plot the length of the wake behind the sphere at a Re=118 is approximately 0.95 of the sphere diameter. The numerical results agree closely with Taneda's flow visualization results.

The wake separation angle for this simulation can be estimated to be around 68. Nakamura [58] studied the wake behind a sphere and presented a curve for variation of the wake separation angle with Reynolds number. For a Re=120 the angle is near

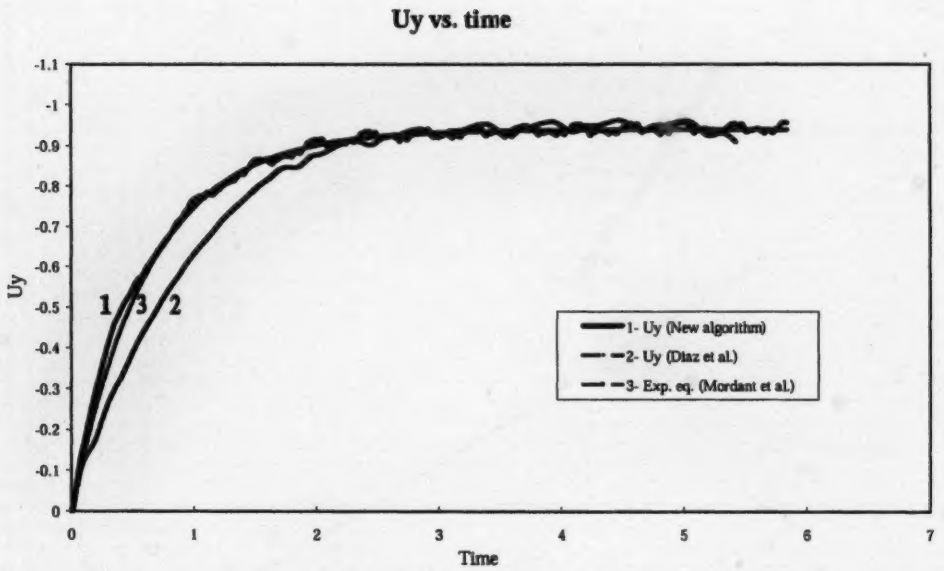

Figure 8.6: Particle falling under gravity. Re=50, Fr=1.12. Velocity vs. time (dimensionless).

60. The larger value we obtained could be attributed to different causes. Nakamura's experiments could not be reproduced exactly since although the experimental setup is described, the exact configuration for a given Reynolds number is not stated (for example the diameter used for the spheres are in a given range). For this reason, not all the parameter could be matched (i.e. Fr number). Other sources of error could be:

- the resolution of the discretized domain (i.e. number of points in the domain mesh) used in this simulation;
- the method used to compute the streamtraces by the visualization software;
- the inaccuracy in determining the the angle of separation from the three-dimensional results.

In the range between $210 < \mathrm{Re} < 270$ the flow becomes non-axisymmetric as the ring vortex shifts off-axis. In contrast to the flow past a circular cylinder, which becomes unsteady upon the loss of symmetry, the double-thread wake observed here remains steady. By Re=270 the double-thread wake is itself observed to become unstable and eventually vortex loops begin shedding from the sphere as so-called hairpin vortices [17]. Tomboulides [72] presented numerical results from a spectral element solution of flow over a sphere for $20 < \mathrm{Re} < 1000$. He showed steady axisymmetry flow for $\mathrm{Re} < 212$. Johnson and Patel [44] implemented a finite difference discretization on a three-dimensional mesh with spherical topology. The method provides a very good accuracy for resolving finely the spatial structures of the flow but it is costly for unsteady simulations. Their simulations were carried out for Re between 50 and 300 and compared to flow visualization results. Johnson and Patel found a critical Reynolds number for axisymmetry breaking at $\mathrm{Re} = 210$. Above this threshold the flow was found to be steady and to have only a planar symmetry. The onset of unsteadiness was reported in the interval of $\mathrm{Re} \in [270, 280]$. Ghidersa et al. [29] studied the primary and secondary instabilities of the sphere wake from the nonlinear dynamic systems theory viewpoint. For the primary bifurcation they provide a theory of axisymmetry breaking by a regular bifurcation. For the secondary bifurcation, basic linear and non linear characteristics are given. The periodic regime is described as a limit cycle. In their work they present proof that the axisymmetry breaks before the Hopf bifurcation sets in and that the axisymmetry gives way to a plane symmetry, the plane being arbitrary.

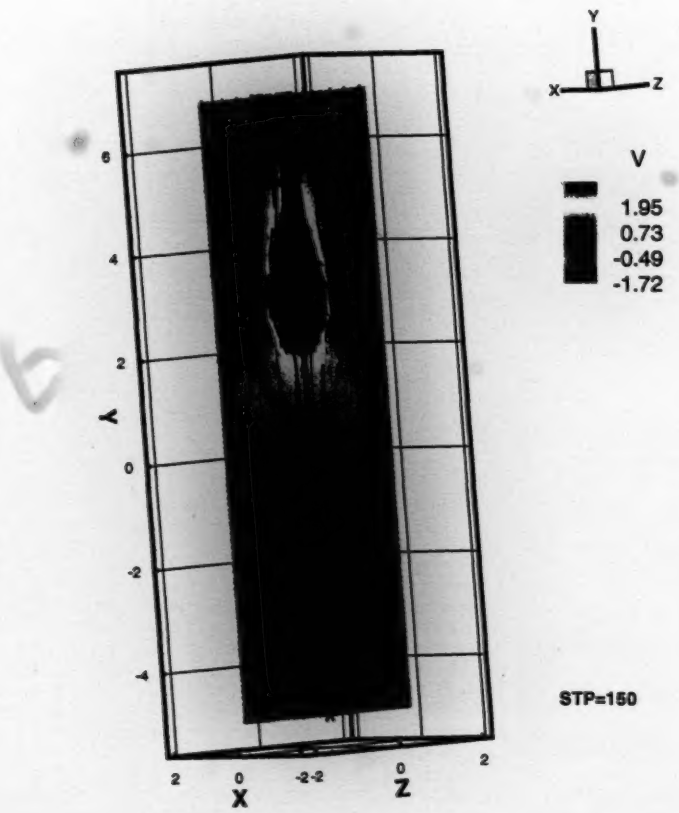

Figure 8.7: One particle settling under gravity. Re=118.0, Fr=0.01. Streamtraces.

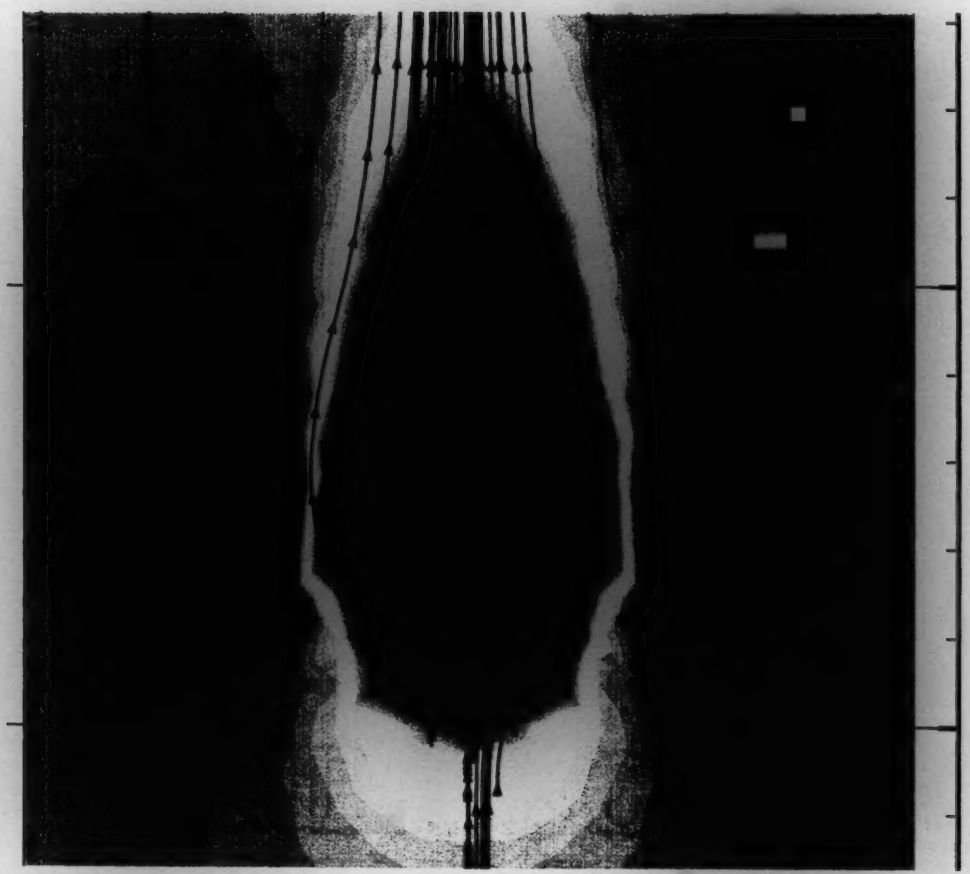

Figure 8.8: One particle settling under gravity. Re=118.0, Fr=0.01. Wake behind the sphere.

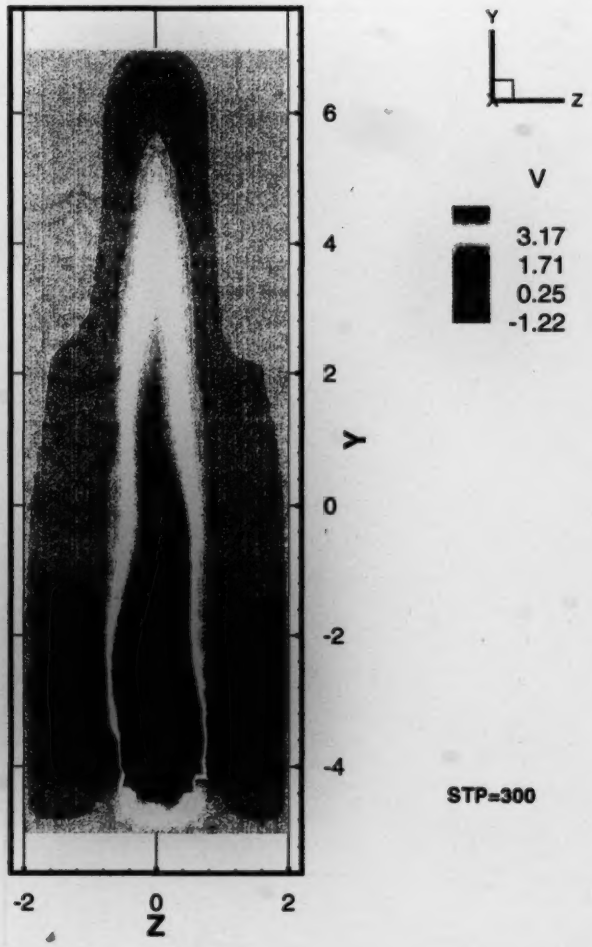

Figure 8.9: One particle settling. Re=220.0, Fr=0.01. Uy contour at x=0 (dimensionless).

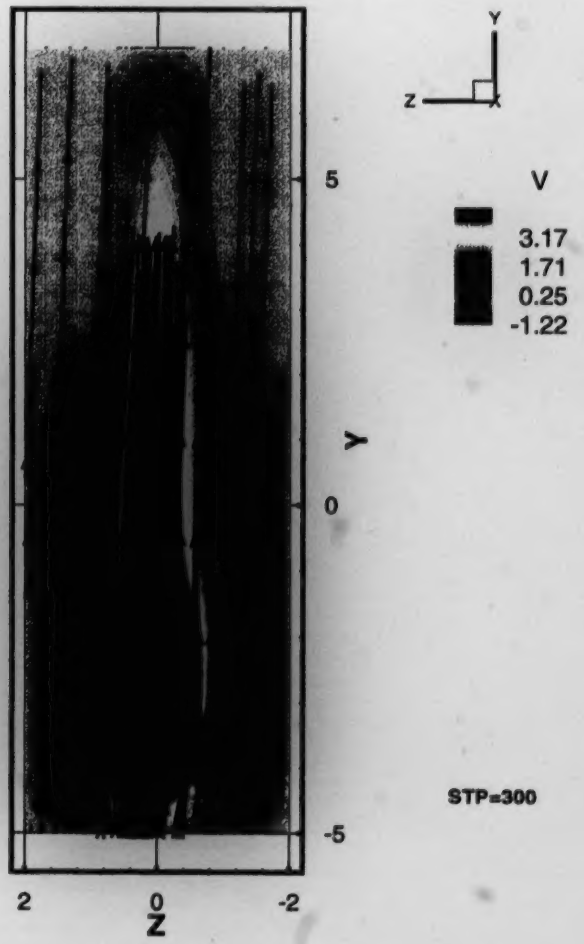

Figure 8.10: One particle settling. Re=220.0, Fr=0.01. Streamtraces.

Figures 8.9 and 8.10 shows the results for a sphere settling with Re=220. The simulation was carried out using a mesh with 306231 nodes and 205800 elements and a timestep of dt=0.001. At this higher Reynolds number the solution no longer exhibits axial symmetry. However, the flow remained steady.

Figures 8.11 shows the results for Re=300 and Fr=0.01. The mesh for this simulation had 306231 nodes and 205800 elements. The timestep was set to dt=0.001.

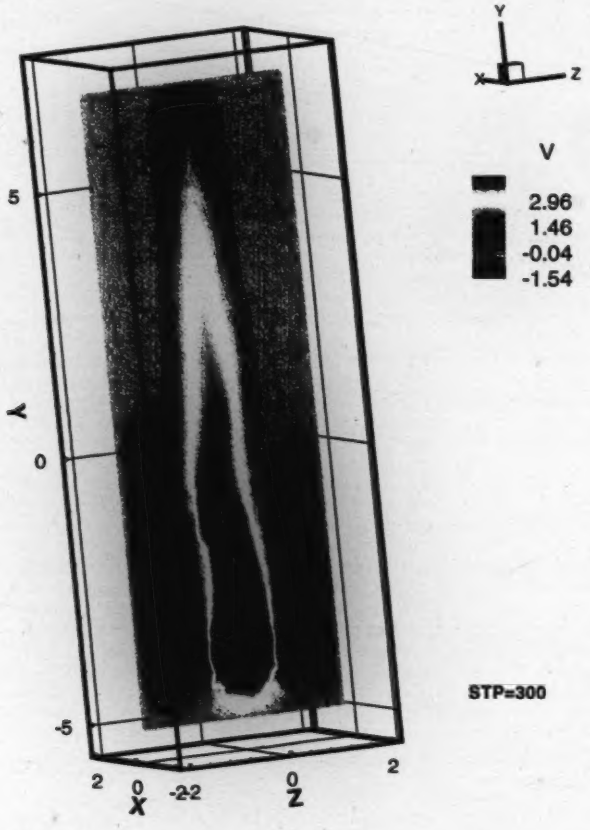

Figure 8.11: One particle settling. Re=300.0, Fr=0.01. Streamtraces.

These results are in agreement with published results from Johnson and Patel [44].

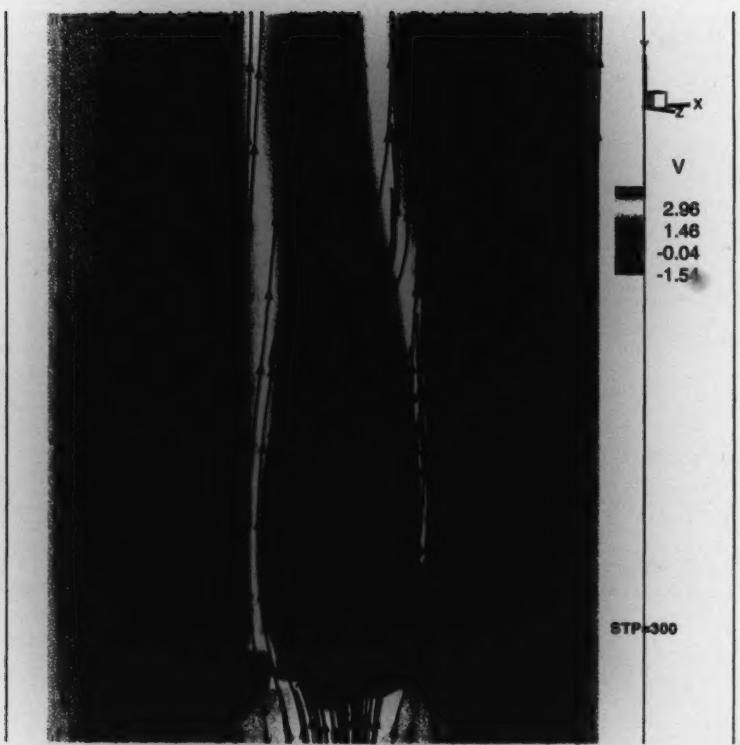

Figure 8.12: One particle settling. Re=300.0, Fr=0.01. Streamtraces.

8.3 One particle in a rotating cylinder

In order to test the angular velocity computations simulations were carried out for a case of one spherical particle inside a cylinder. The mesh for this case had 63679 nodes and 44362 elements, and the time step was dt=0.01. At the beginning of the simulation the fluid was at rest and the cylinder's walls rotate with an angular velocity of $\Omega_z = 0.01$.

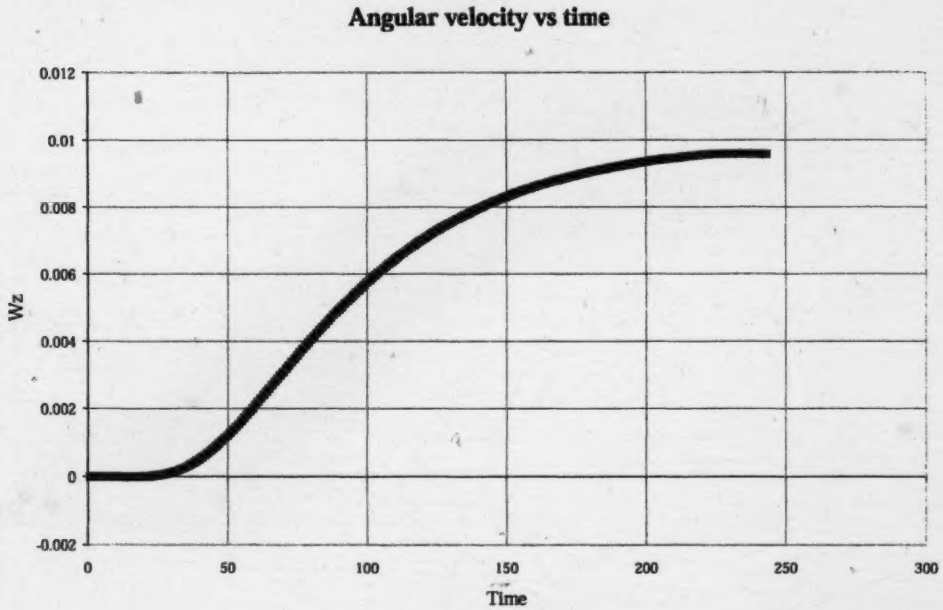

Figure 8.13: Sphere in a rotating cylinder: Angular velocity vs. timestep (dimensionless).

Figures 8.15-8.17 show the rotating cylinder and the streamtraces at a plane passing through the center of the sphere for different timestep. Figure 8.13 shows the evolution of the sphere's angular velocity with time. The angular velocity increases until it almost matches that of the wall. The difference between the angular velocities (sphere and walls) can be attributed to the mesh resolution.

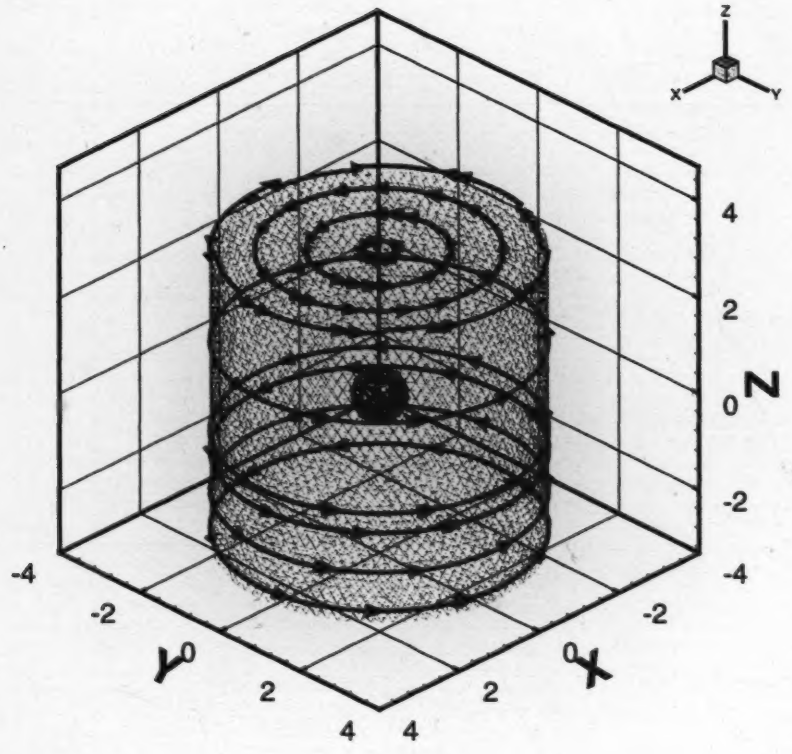

Figure 8.14: Sphere in a rotating cylinder, computational domain.

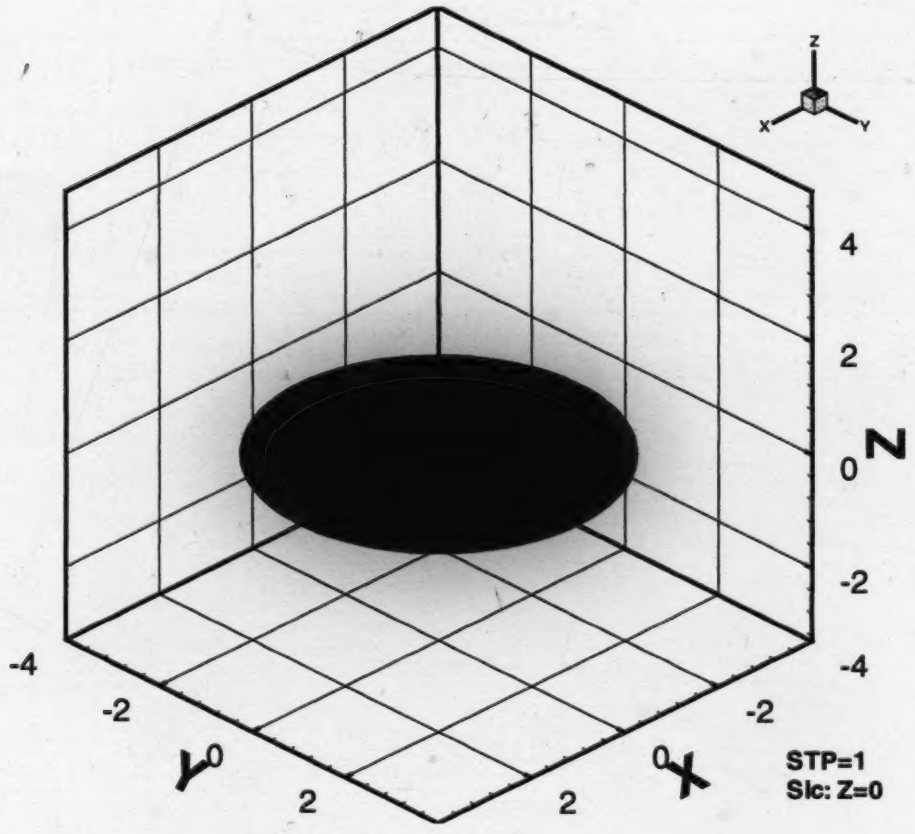

Figure 8.15: Sphere in a rotating cylinder, t=0.1 (dimensionless).

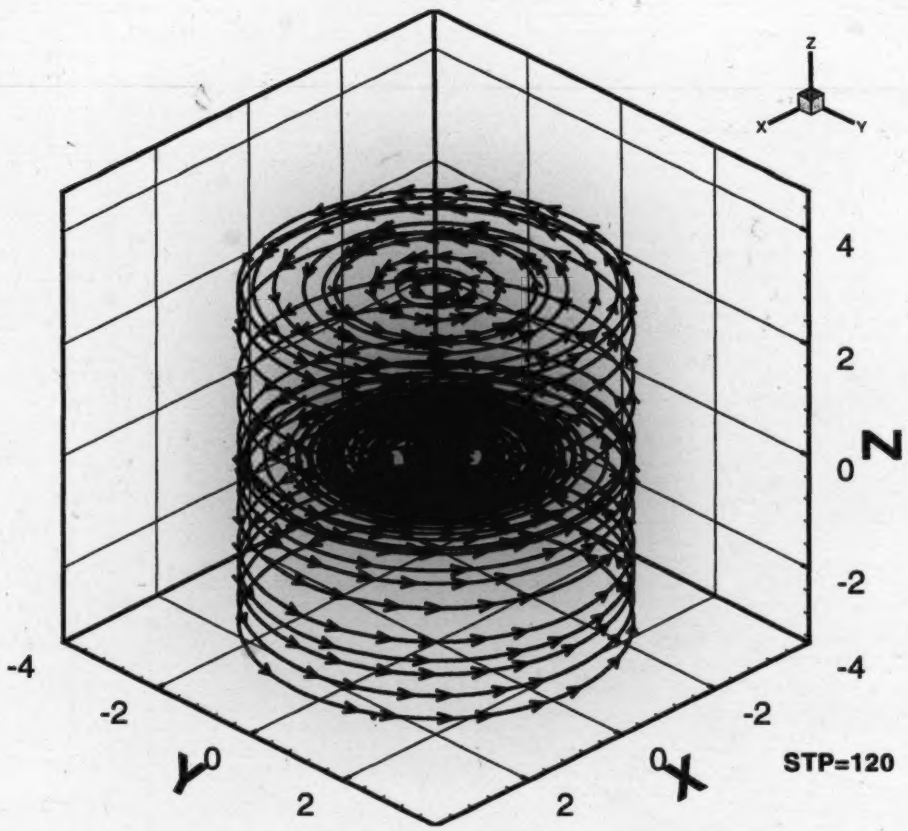

Figure 8.16: Sphere in a rotating cylinder, t=1.2 (dimensionless).

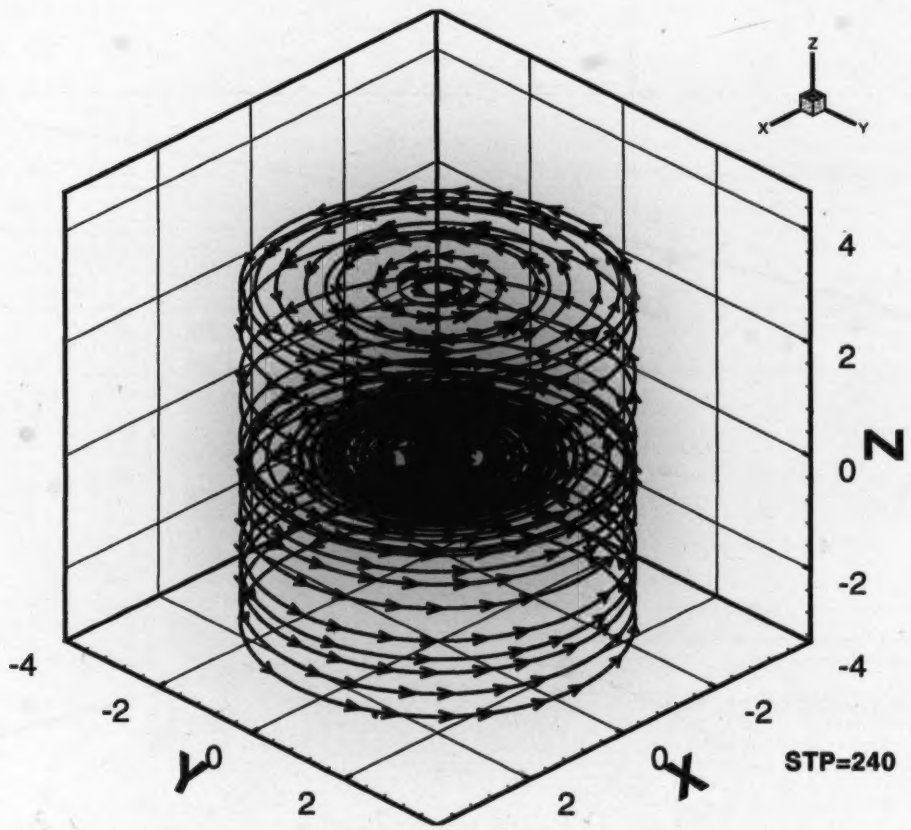

Figure 8.17: Sphere in a rotating cylinder, t=2.4 (dimensionless).

Experiment	X_0 from wall (cm)	ΔX traveled (cm)	ΔX (diameters)	Uy_{inf}(cm/s)
1	0.055	0.147	0.739	30.2
2	0.086	0.171	0.860	31.2
3	0.220	0.091	0.457	30.6
4	1.938	0	0	30.6

Table 8.1: Wall effects: initial position, horizontal distance traveled and terminal velocity.

8.4 Wall effects on the terminal velocity

It is well known that when a sphere is dropped near a vertical wall in a Newtonian liquid it moves away from the wall as it falls downward under gravity [67]. The problem of interaction between a sphere and a wall was first studied by Goldman, Cox and Bremmer [35] who find that a sphere falling in a channel filled with a liquid cannot be in contact with the vertical walls, as a lubrication layer develops between the wall and the falling sphere. Their study also showed that under certain conditions the sphere falls downward while rotating in a clockwise direction as if the sphere was rolling up the wall. This sense of rotation is referred as anomalous as opposed to a normal rotation when the sphere rotates down an inclined plane without slipping.

Wall effects are examined and numerical and experimentally results are compared qualitatively. Figure 8.18 shows the images obtained experimentally for a particle that is released closed to the wall.

Figures 8.19 through 8.22 show the vertical velocity versus time and the particle's displacement versus time for four drops at a varying distance from the wall. From these plots we can see that the particle moves away from the wall towards the center of the tank. For each case the initial distance from the wall, the distance travelled and the terminal velocity are summarized in Table 8.1.

From table 8.1 we see that in experiment 2 the sphere traveled a longer distance from the wall than in experiment 1 although in the former case it was released further away from the wall than in the latter. In comparing these results we should note that this could have been caused by the releasing technique on the sphere or on the error in analyzing the images.

To evaluate the wall effects on our numerical simulations two different situations were tested: a particle released from a starting position of (x,y,z) = (1.6, 6, 0) giving

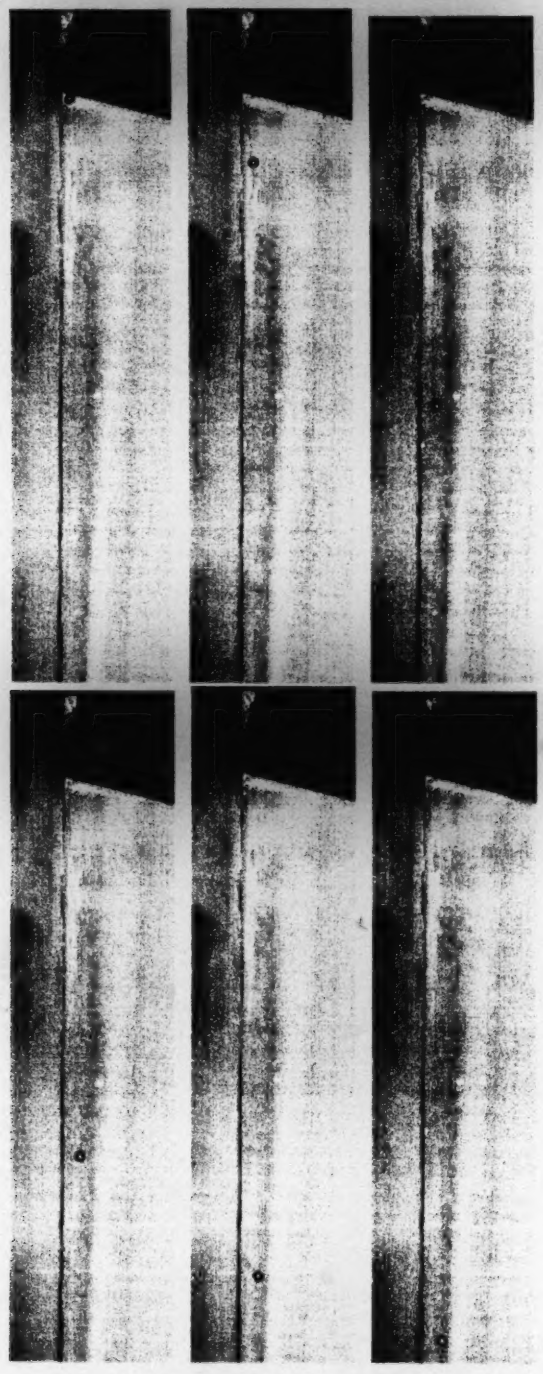

Figure 8.18: Wall effects: experimental results.

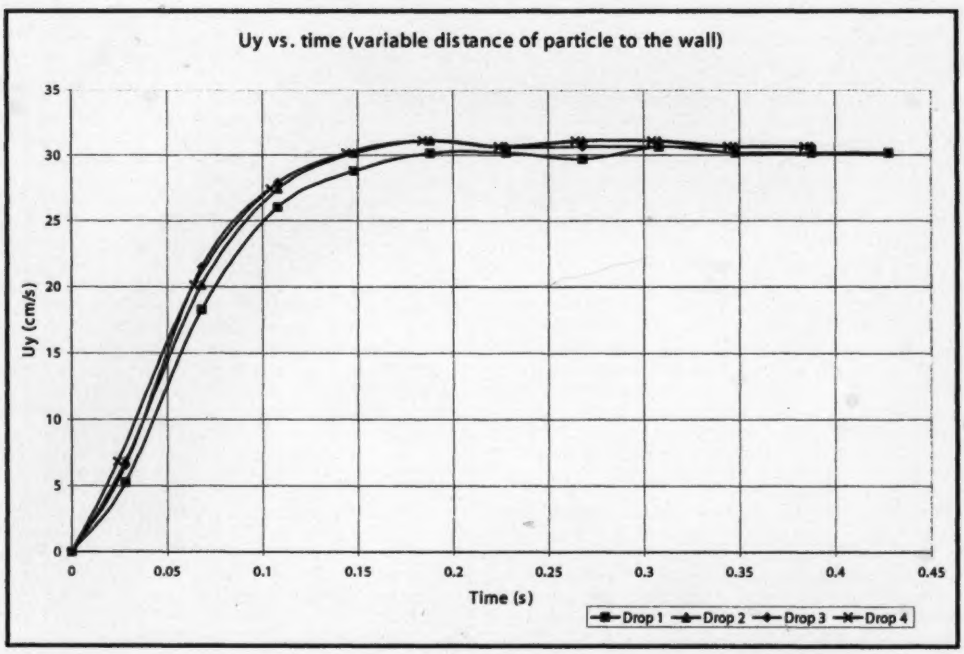

Figure 8.19: Wall effects: experimental results. Vertical velocity vs. time.

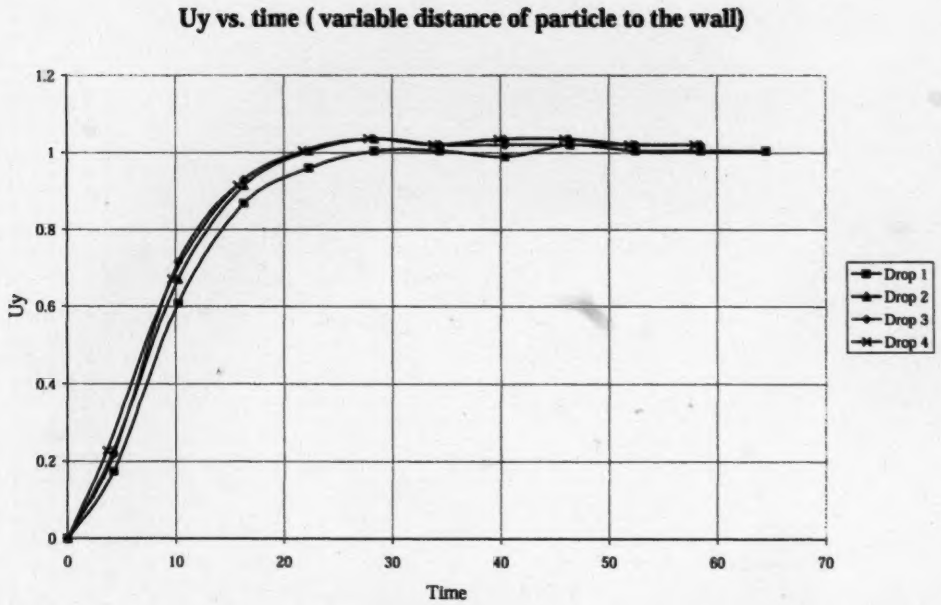

Figure 8.20: Wall effects: experimental results. Vertical velocity vs. time (dimensionless).

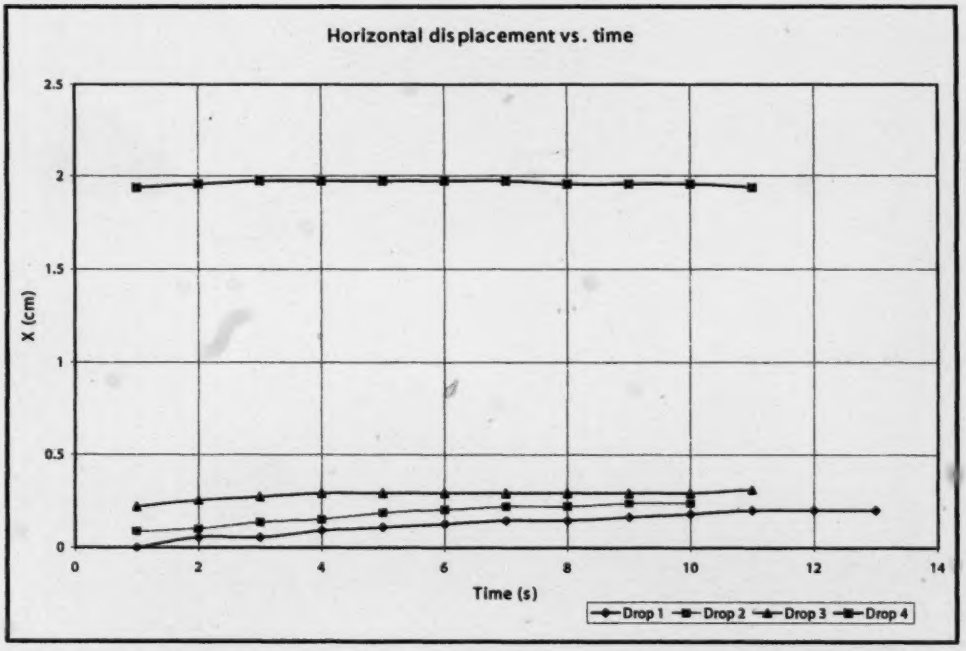

Figure 8.21: Wall effects: experimental results. Horizontal displacement vs. time.

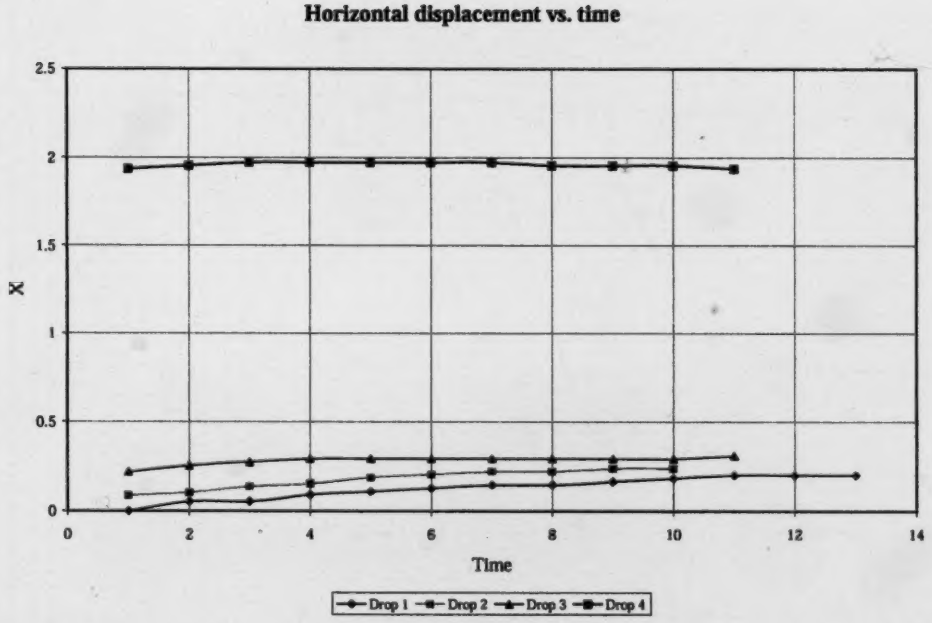

Figure 8.22: Wall effects: experimental results. Horizontal displacement vs. time (dimensionless).

a distance of 0.1 from the wall and a second case where the particle is released from a starting position of (1.3, 6, 0) with a distance of 0.3 from the wall. For these cases the meshed used had 306231 nodes and 205800 elements and the timestep was 0.01. Other parameters used were: Re=100.0 and Fr=0.1 and a ratio of densities of 2.56.

Figures 8.23 and 8.26 show the numerical simulation results.

As in the experimental case the particle settles moving away from the wall till it reaches a stable position with respect to the wall of the channel. In both cases the horizontal displacement away from the wall finishes when the center of the sphere is at x=0.9 and the separation to wall just over one radius of the sphere. The terminal velocity reached is almost the same in both cases. The numerical results show the sphere exhibits a small oscillation in the z direction as well as rotation with respect to the z axis. The rotation is anomalous at first, when the sphere is close to the wall. The maximum angular velocity was found to be $\omega_z = -0.42$. As the spheres moves to the middle of the channel the rotations changes direction.

The findings regarding the rotation of the sphere could not be confirmed through our experiments since we were not able to measure rotations and the very small oscillations would be in the order of magnitude of the error of our measures. However, we could explain these results by analyzing the velocity and pressure around the sphere. The velocity on the particle surface on the side that is closer to the middle of the channel is greater than the velocity on the side next to the wall. Because of this, the pressure on the side next to center is smaller than the pressure on the side next to the wall. Because of this pressure difference there is a net pressure force acting on the sphere in the direction away from the wall. As the sphere falls downward the fluid has to go around it, and since the velocity is different on each side of the sphere, it rotates in an anomalous manner.

Singh et al. [67] studied the sedimentation of a sphere near a vertical wall in an Oldroyd-B fluid. They also presented some results of a particle dropping in a Newtonian fluid. Their study showed that the particles drifted away from the wall with an anomalous rotation till it reached a steady state position somewhere between the channel wall and the channel center. When the particle was away from the wall it rotated in normal manner. Our simulations reproduce these features qualitatively.

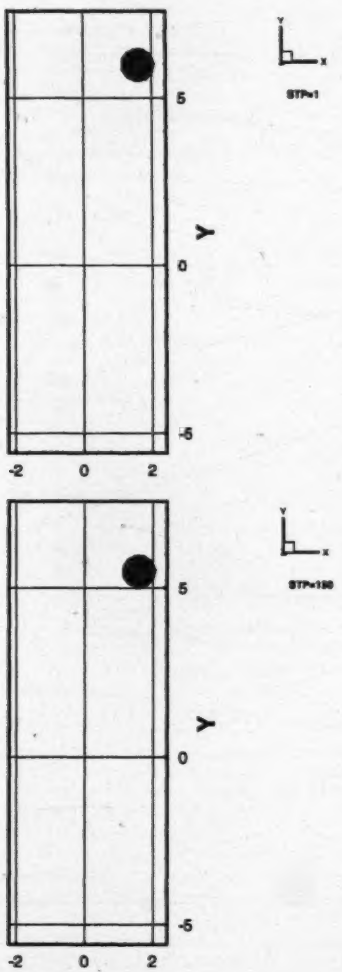

Figure 8.23: Wall effects: numerical results. Case 1- Initial distance to the wall=0.1 (dimensionless). t=1dt and 150dt.

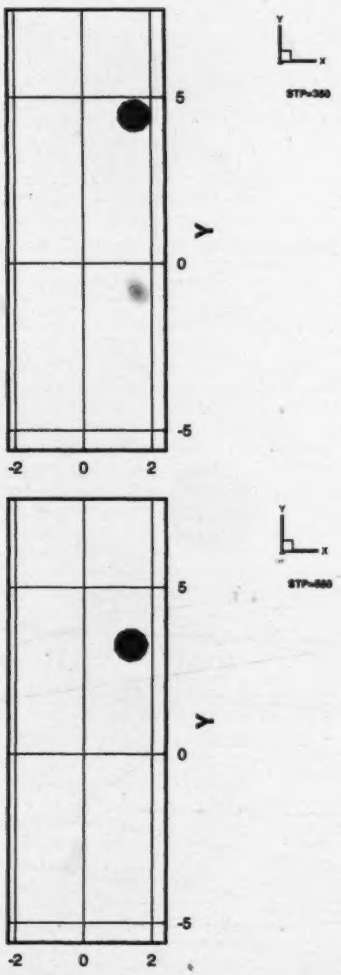

Figure 8.24: Wall effects: numerical results. Case 1- Initial distance to the wall=0.1 (dimensionless). t=350dt and 550dt.

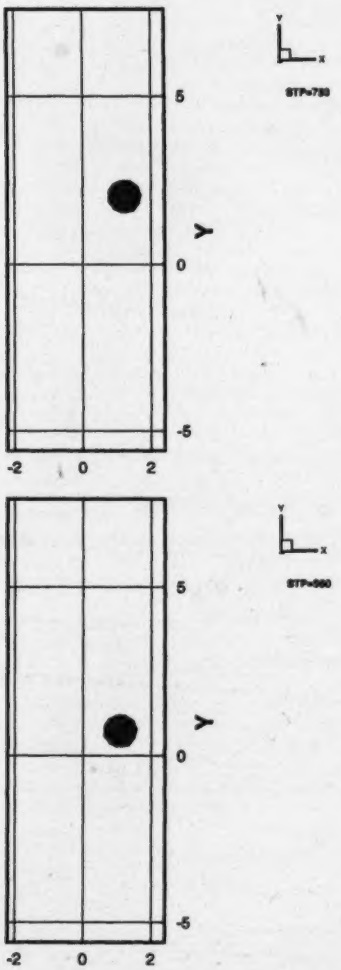

Figure 8.25: Wall effects: numerical results. Case 1- Initial distance to the wall=0.1 (dimensionless). t=750dt and 950dt.

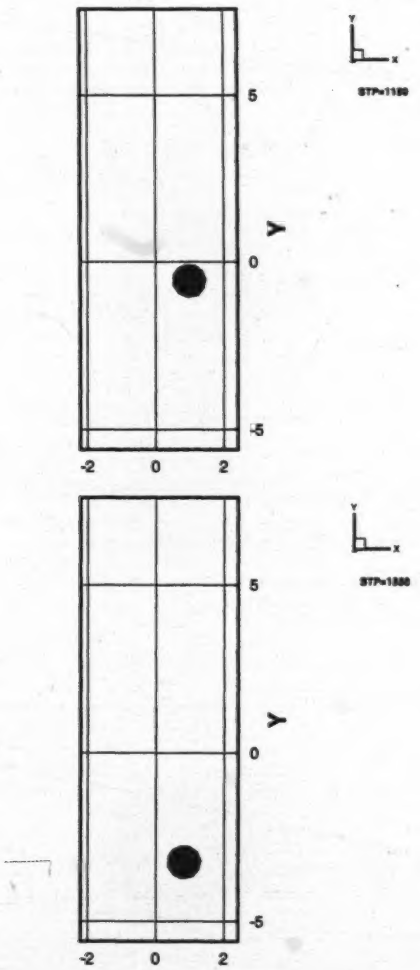

Figure 8.26: Wall effects: numerical results. Case 1- Initial distance to the wall=0.1 (dimensionless). t=1150dt and 1550dt.

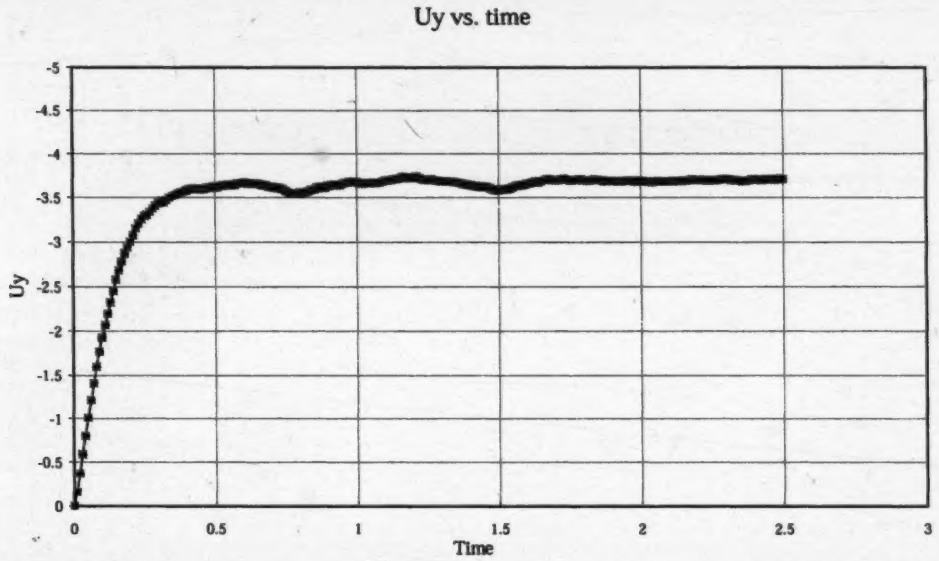

Figure 8.27: Numerical results. Vertical velocity vs. time (dimensionless). Particle released at 0.1 (dimensionless) from the wall.

8.5 Collision with walls

Collisions were first studied experimentally. Figures 8.35 to 8.39 show the results obtained. For different kind of sphere (steel bearing and nylon spheres) the behavior was quite different. In the case of the steel bearing the sphere will bounce once or twice and then stop. On the other hand, the nylon particle came to rest as soon as it touched the bottom wall.

The presence of a lubricant has a significant influence on the dynamics of colliding bodies. A detailed analysis of these close contact motions requires consideration of the dynamic shape and separation of the particles surfaces. Molecular and hydrodynamic forces acting upon nearly touching surfaces can cause the particle to deform unless they are very rigid. This deformation is caused by normal forces distributed over the surface of an elastic solid and it can be modeled by integrating the surfaces-stress distribution. The deformed shape can be determined from the pressure profile in the fluid layer between the solid surfaces. Davis et al. [20] studied collisions between two spheres and between a sphere and a wall. They explained the deformation and collision by analyzing the pressure changes. When the sphere is about to collide

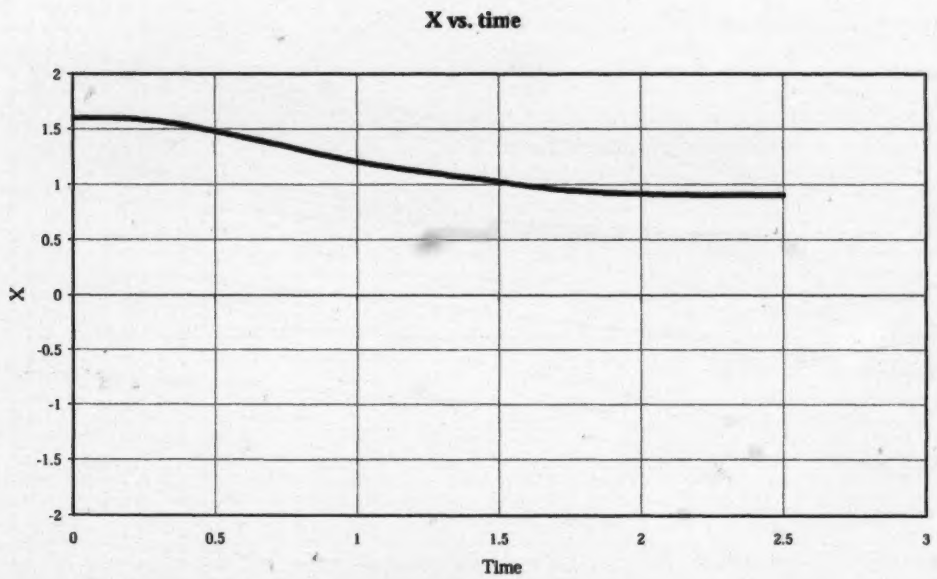

Figure 8.28: Numerical results. Horizontal displacement vs. time (dimensionless). Particle released at 0.1 (dimensionless) from the wall.

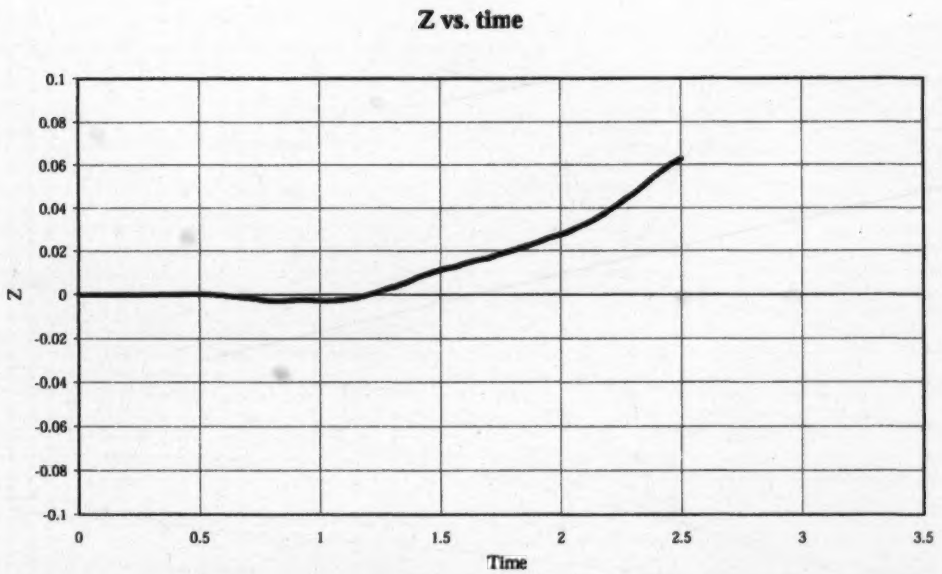

Figure 8.29: Numerical results. Displacement in Z vs. time (dimensionless). Particle released 0.1 (dimensionless) from the wall.

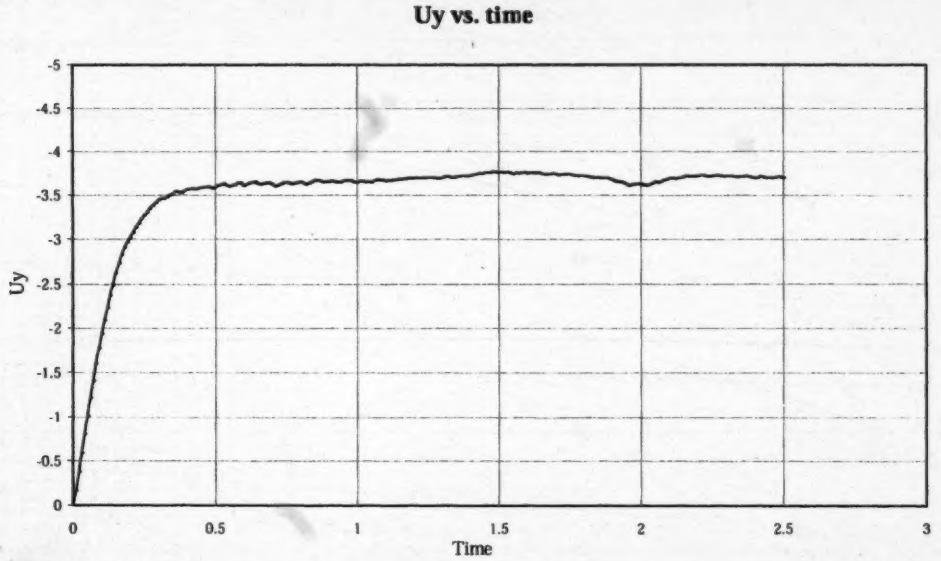

Figure 8.30: Numerical results. Vertical velocity vs. time (dimensionless). Particle released 0.3 (dimensionless) from the wall.

the gap between the surfaces decreases and the pressure increases, this pressure may deform the particle. However, the pressure also causes the sphere to slow down. The deformation reaches a maximum and then relaxation occurs. The distance of the rebound is limited by the viscous dissipation occurring as the sphere recedes and the fluid flows back into the gap under suction. Eventually, the non-conservative viscous forces damp out the motion.

Barnocky et al.[7] studied the dynamic deformation of a solid elastic sphere immersed in a viscous fluid and in close motion toward another sphere or a plane solid surface. Their work provided a rational criteria for prediction whether a solid particle will stick or rebound subsequent to impact. If there is a rebound, the hydrodynamic pressure which builds up in the thin layer must become large enough to elastically deform the sphere. Under these extremes pressures, it is expected that the fluid may also compress and that its viscosity may increase by several order of magnitudes. An increase in the fluid viscosity may alter the collision dynamics by increasing the hydrodynamic pressure and viscous dissipation. Barnocky's study makes quantitative prediction of the influences of the pressure-dependent fluid properties of density and

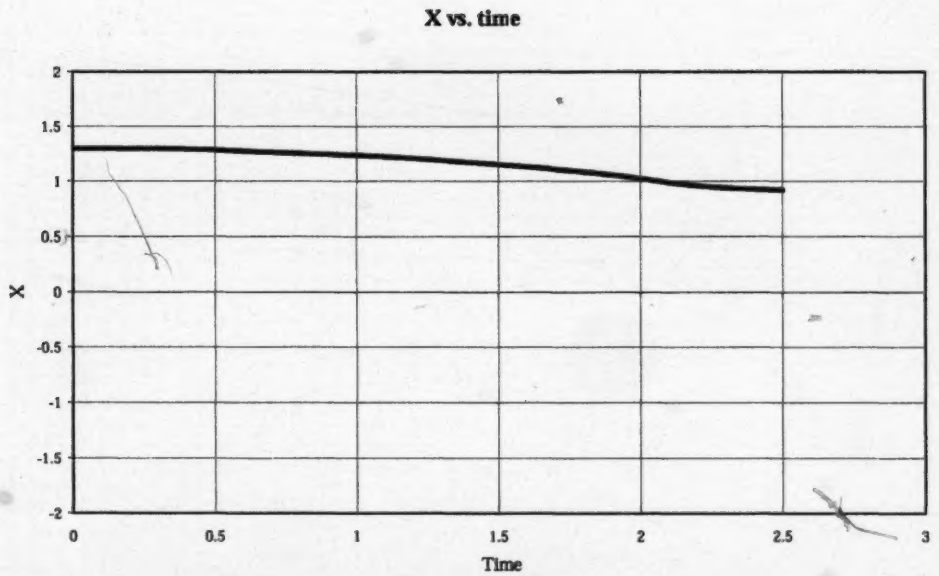

Figure 8.31: Numerical results. Horizontal displacement vs. time (dimensionless). Particle released 0.3 (dimensionless) from the wall.

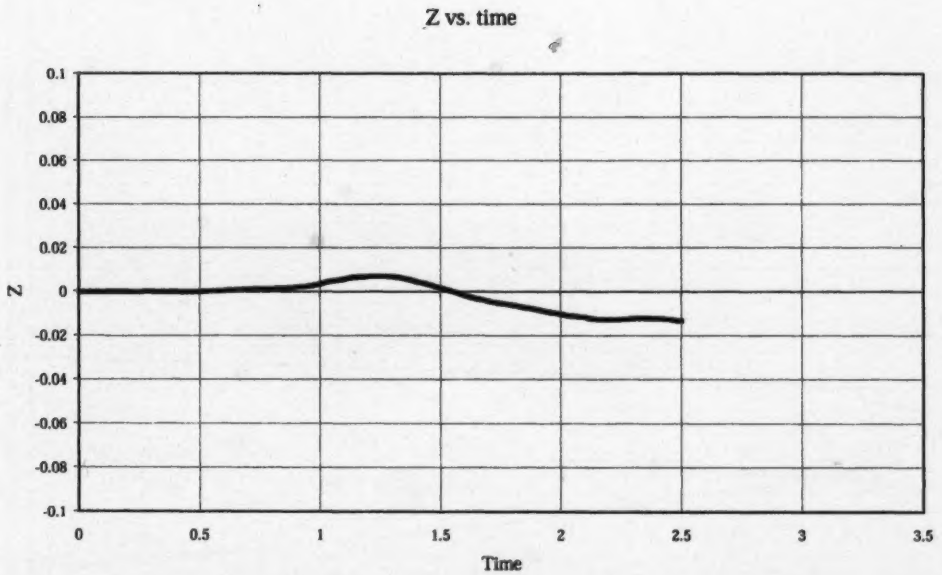

Figure 8.32: Numerical results. Displacement is Z vs. time (dimensionless). Particle released 0.3 (dimensionless) from the wall.

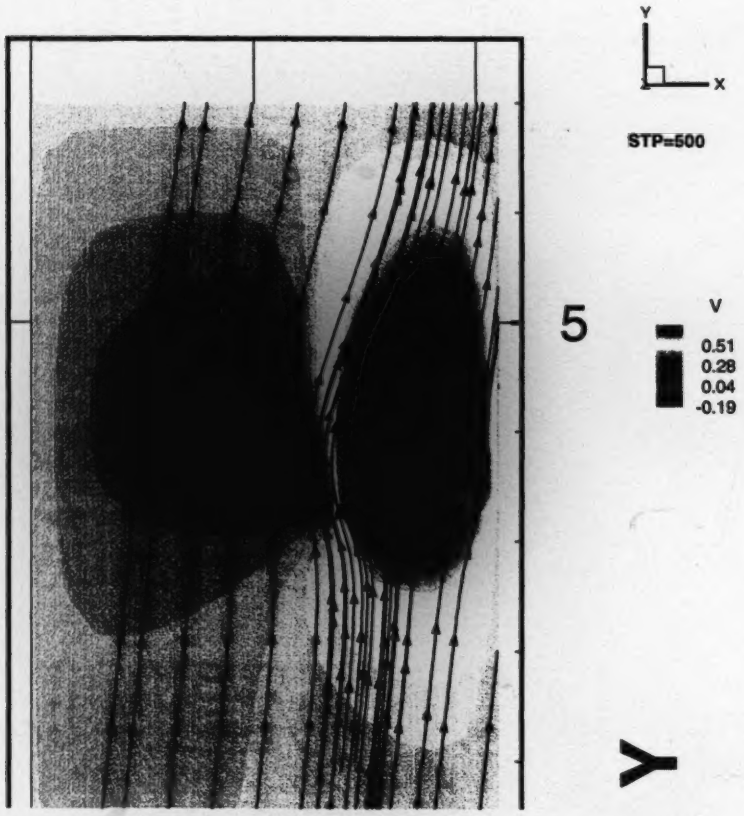

Figure 8.33: Wall effects: numerical results. Streamtraces for t=500dt.

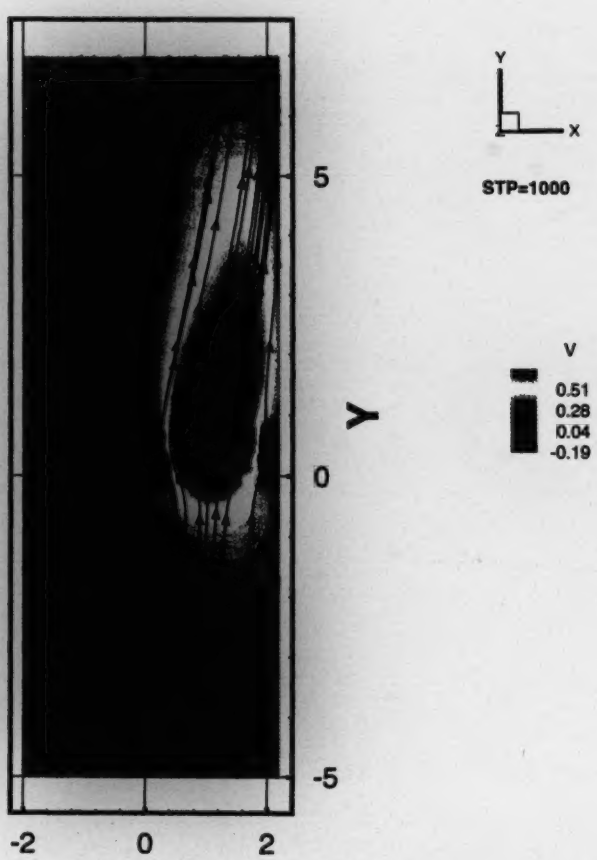

Figure 8.34: Wall effects: numerical results. Streamtraces for t=1000dt.

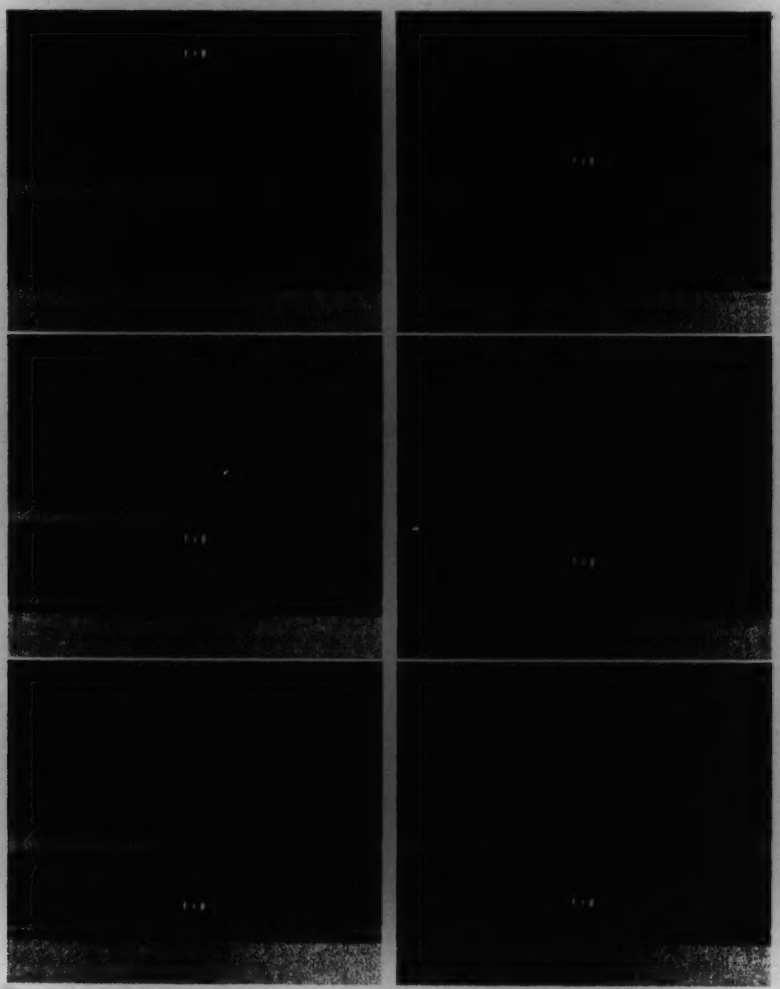

Figure 8.35: Wall collision: experimental results. t=0.04s, 0.08s, 0.10s, 0.108s, 0.116s and 0.12s.

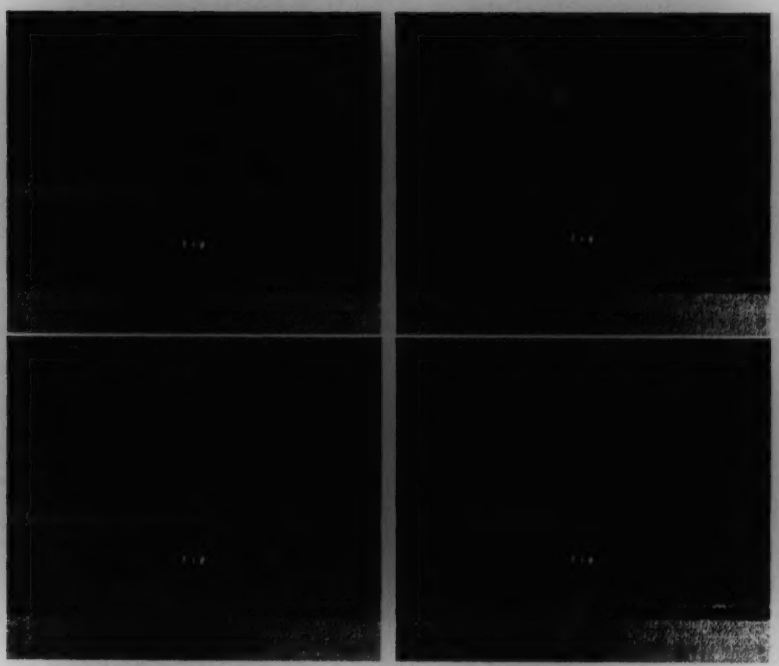

Figure 8.36: Wall collision: experimental results. t=0.124s, 0.132s, 0.14s and 0.148s.

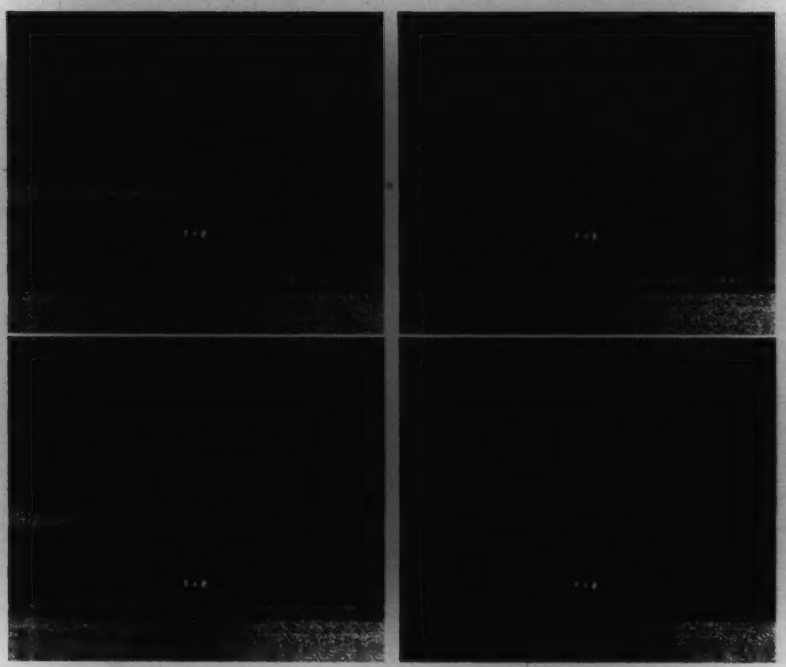

Figure 8.37: Wall collision: experimental results. t=0.156s, 0.164s, 0.20s, 0.24s.

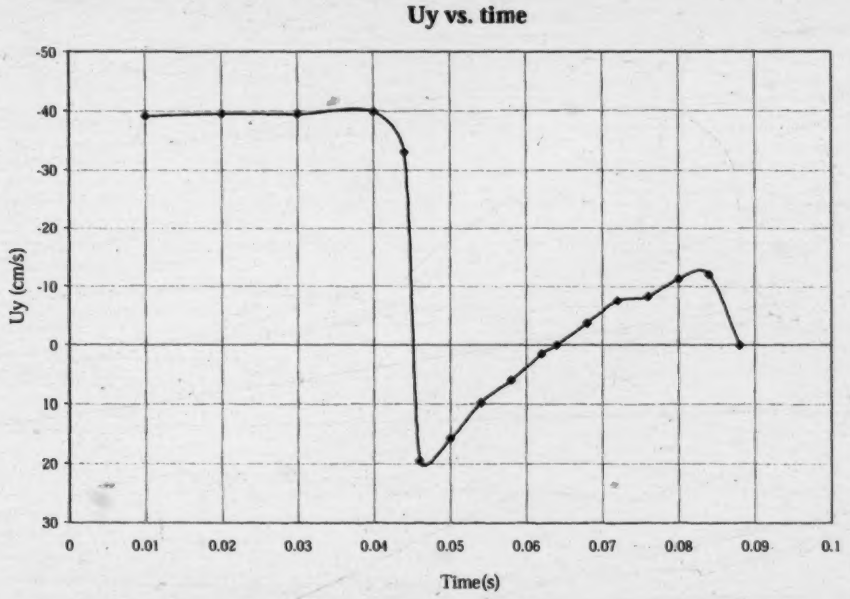

Figure 8.38: Wall collision: Vertical velocity vs. time for a steel bearing.

viscosity on the collision of spheres in a viscous fluid.

Since the current thesis was not focused on collisions, a simple approach was implemented to handle these situations. Its goal was to avoid particles from penetrating the walls without decreasing the code's performance. To deal with wall collision several strategies were tried. The first approach was to reduce the time step when the particle was near a wall in order to prevent it from penetrating the walls. In this case, because the Lagrange multipliers are defined globally they would cause the particle to slow its motion, the particle will come to rest when the distance to the wall was small enough. However, no rebound could be modeled, and for some cases the constrain on the time step would be too severe. The second approach to deal with collisions between a particle and the wall was to simulate an elastic collision where the sphere will basically bounce back. Although this approach avoided particle from penetrating the wall, caused the particle to bounce several times before going to rest. The next approach was to modify the previous method by introducing a dampening coefficient to make the particle's bouncing decay faster. Even though this last approach avoids the particle from penetrating the wall, the dampening coefficient had to be tuned for each specific case. A more general approach, based on pressure-dependent density

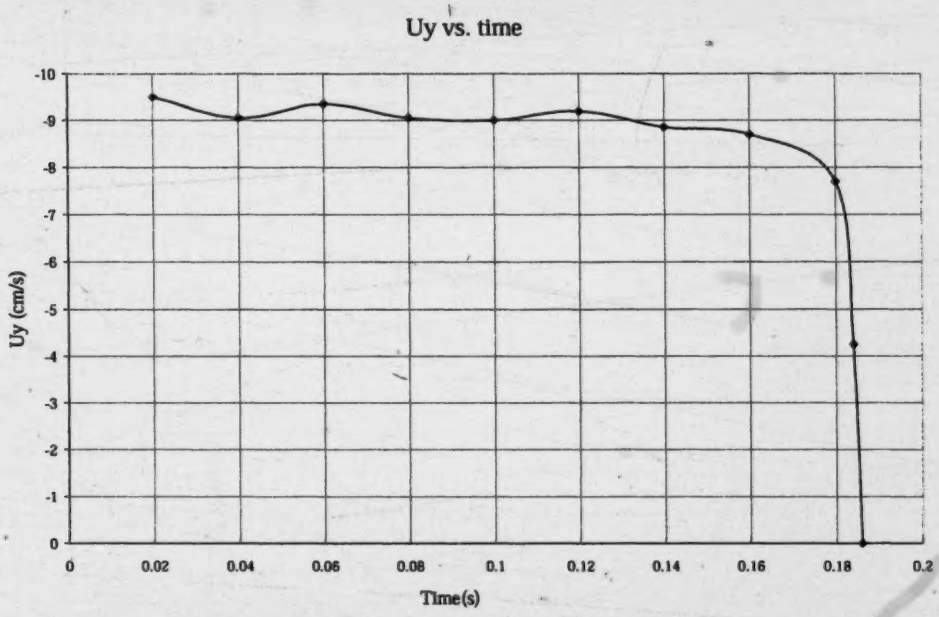

Figure 8.39: Wall collision: Vertical velocity vs. time for a nylon sphere.

and viscosity on this hydrodynamic collisions should be developed in the future. Such an approach would require mesh refinement and should not only follow the motion along the line of center of mass of the sphere but also take into account tangential component of particle collisions.

Chapter 9

Numerical and experimental results: multiple particle modeling

9.1 Two sphere settling in horizontal configuration

The code was tried with more than one solid particle. Here we present the case of two solid particles that are at the same initial vertical position. Two steel spheres were released in a fluid at rest. The diameter of the particles was 1.99mm. The Reynolds number based on the terminal velocity of one of the particle is approximately Re=29.8 and the Froude number is Fr= 5.3. Under these conditions both particles settle side by side of each other without any interaction between them. The terminal vertical velocity reached by each particle is approximately the same: 33 cm/s. The particles exhibit small oscillations in their velocities in the other two components (Ux and Uz).

Figures 9.1-9.3 show the experimental results. Figure 9.4 shows the particles' vertical velocity versus time. In this plot P1 and P2 refer to the sphere that is initially to the left and right in the images obtained and analyzed in our laboratory (Figures 9.1-9.3).

This experiment was reproduced with our numerical code. For the code the mesh had 179401 nodes and 120000 elements. As before, all variables were made dimensionless using a characteristic velocity (terminal velocity) and a characteristic length and introducing the Reynolds and Froude number. The particle initial positions were $(x, y, z)_1 = (-0.7, 6.0, 0.0)$ and $(x, y, z)_2 = (0.7, 6.0, 0.0)$. and the timestep dt=0.01. Figures 9.5 and 9.5 show the particle's position for different timesteps.

In our numerical experiment the spheres have the same curve of vertical velocity

Figure 9.1: Two spheres in horizontal configuration. t=0s, 0.08s, 0.12s.

Figure 9.2: Two spheres in horizontal configuration. t=0.16s, 0.24s, 0.32s.

Figure 9.3: Two spheres in horizontal configuration. t=0.40s, 0.48s, 0.56s.

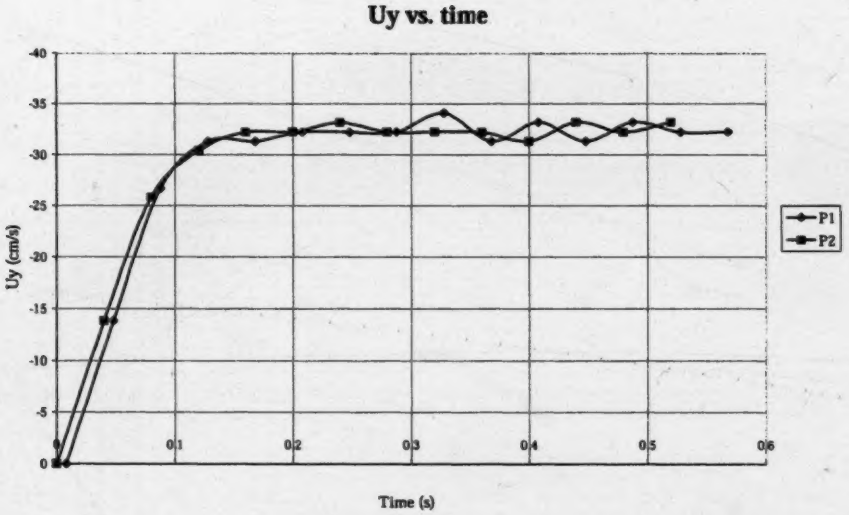

Figure 9.4: Two spheres in horizontal configuration. Experimental results: Vertical velocity vs. time.

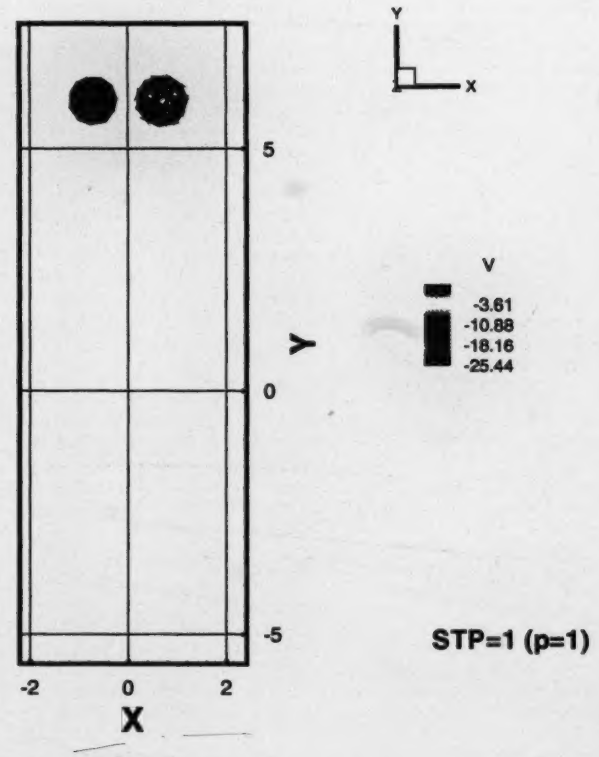

Figure 9.5: Two spheres in horizontal configuration. Numerical results at t=0.01 (dimensionless).

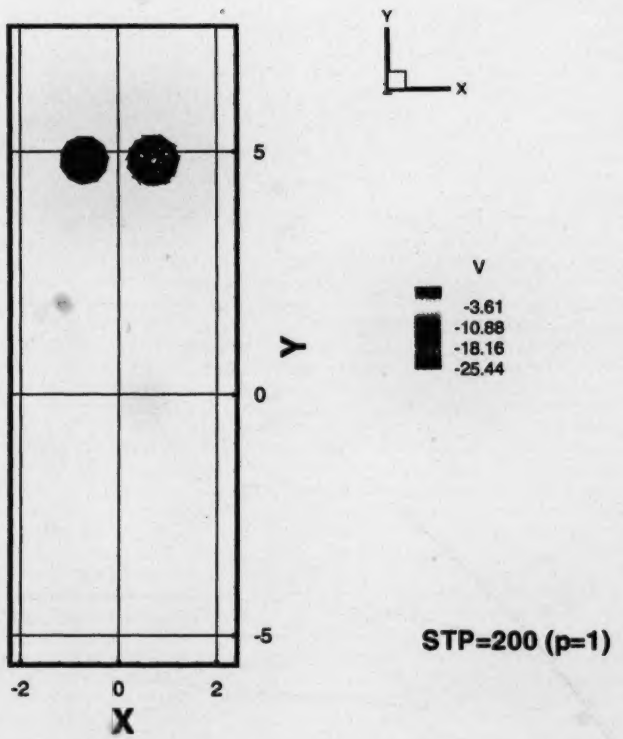

Figure 9.6: Two spheres in horizontal configuration. Numerical results at t=2.0 (dimensionless).

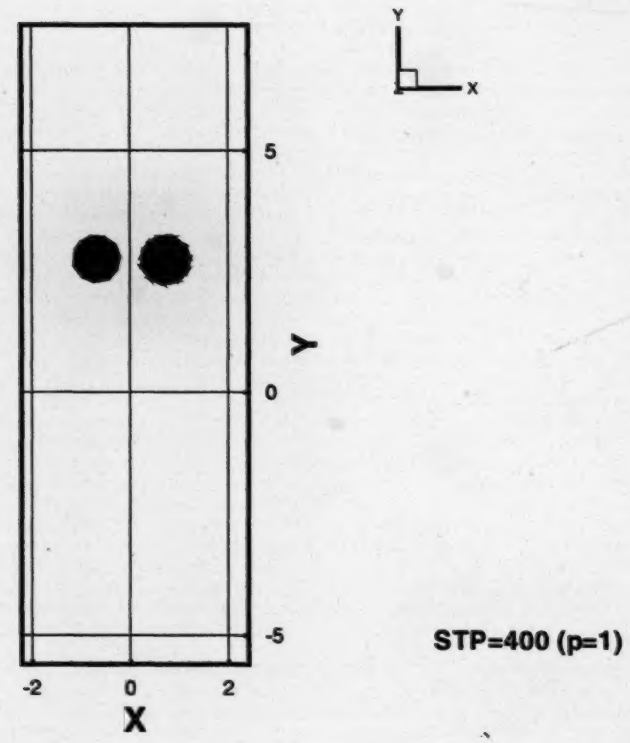

Figure 9.7: Two spheres in horizontal configuration. Numerical results at t=4.0 (dimensionless).

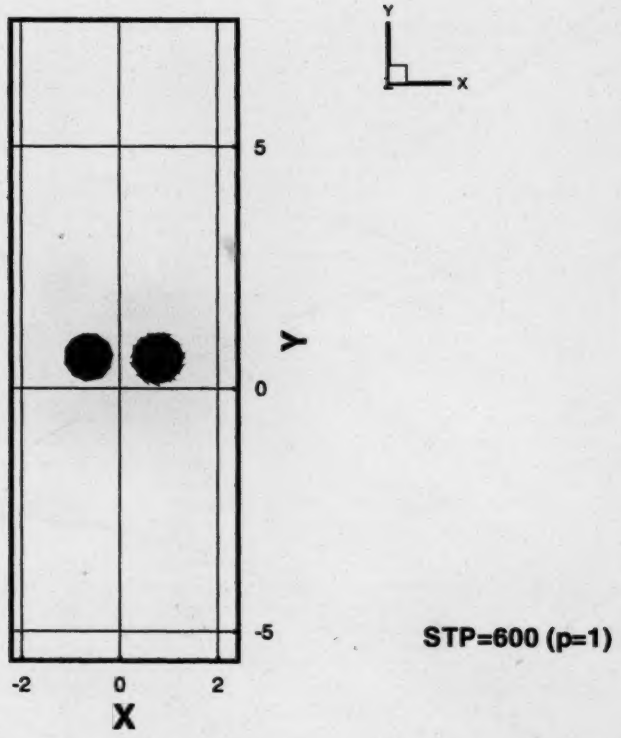

Figure 9.8: Two spheres in horizontal configuration. Numerical results at t=6.0 (dimensionless).

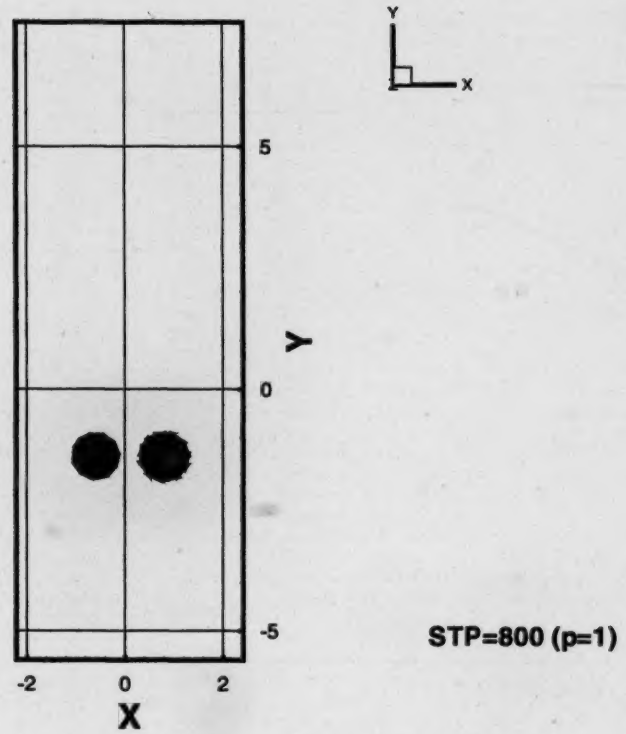

Figure 9.9: Two spheres in horizontal configuration. Numerical results at t=8.0 (dimensionless).

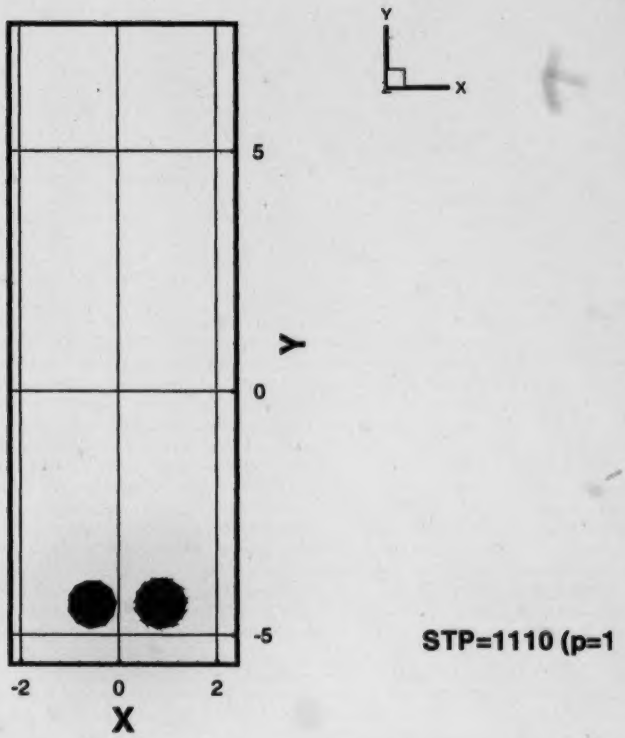

Figure 9.10: Two spheres in horizontal configuration. Numerical results at t=11.0 (dimensionless).

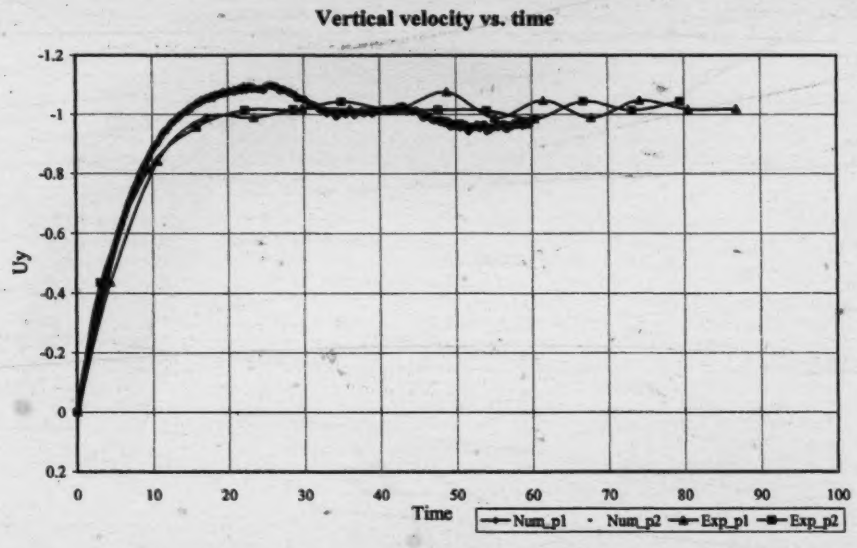

Figure 9.11: Two spheres in horizontal configuration. Experimental and numerical results: Vertical velocity vs. time (dimensionless).

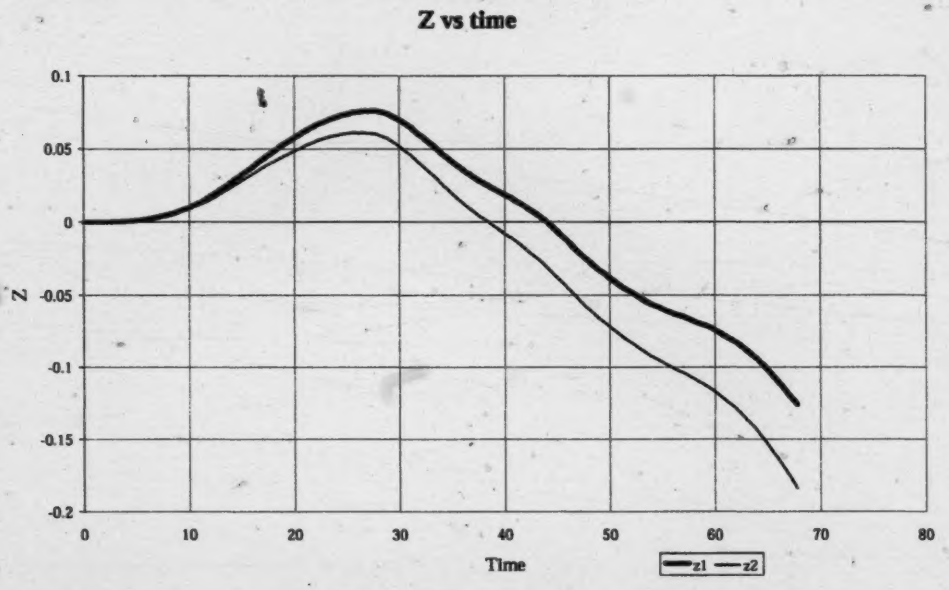

Figure 9.12: Two spheres in horizontal configuration. Numerical results: Z vs. time (dimensionless).

vs. time. The particles also exhibit small oscillations with respect to the x and z axis (Figure 9.12). The numerical results agree with our experimental findings.

9.2 Two sphere settling in vertical configuration

In this simulation two spheres of diameter=1 (dimensionless) were placed one on top of each other and separated by a distance of 1.5 radius (see Figure 9.13). The particle initial positions were $(x, y, z)_1 = (0.0, 6.0, 0.0)$ and $(x, y, z)_2 = (0.0, 4.7, 0.0)$. The ratio of densities was 2.56 and the timestep dt=0.01. The mesh was a rectangular channel of $4 \times 4 \times 12$ and it had 179401 nodes and 120000 elements. The Reynolds number was set to Re=41 and the Froude number to 2.21.

For this case the sphere settled without interaction between each other. The simulation could not be compared to experiments since we were not able to release the particles in this configuration in a reliable way (without disturbing the fluid in the experimental tank and being able to repeat the experiment). Several reasons could explain why the upper sphere did not touch the lower one. One possible explanation could be that for this Reynolds and Froude number (parameters of this simulation) the wake formed behind the particle is too small as to be felt by the sphere on top. According to Taneda's [68] data, the length of the wake behind a sphere settling with a Reynolds number of 40 is around 0.3 of the diameter. Another reason could be that the mesh resolution was not good enough as to reproduce the wake formed.

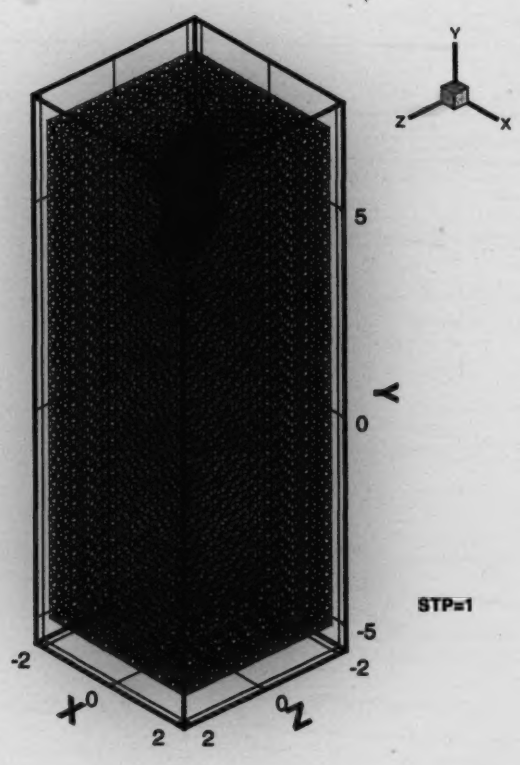

Figure 9.13: Two spheres in vertical configuration. Numerical results at t=0.01 (dimensionless).

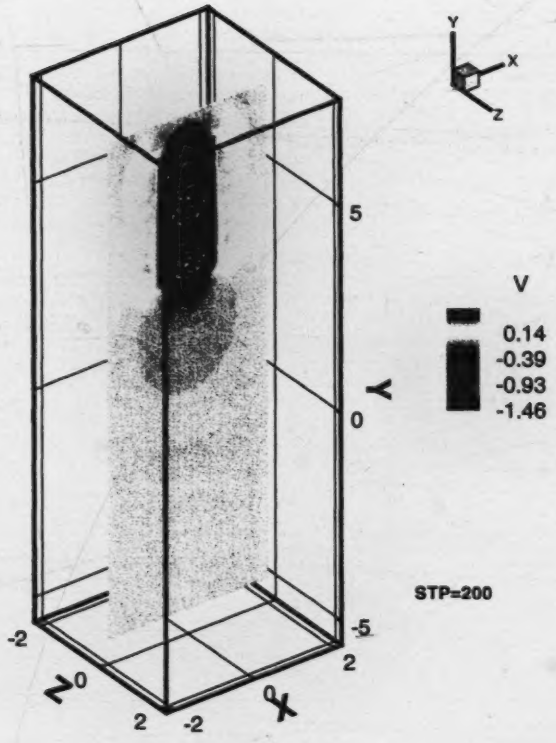

Figure 9.14: Two spheres in vertical configuration. Numerical results at t=2.0 (dimensionless).

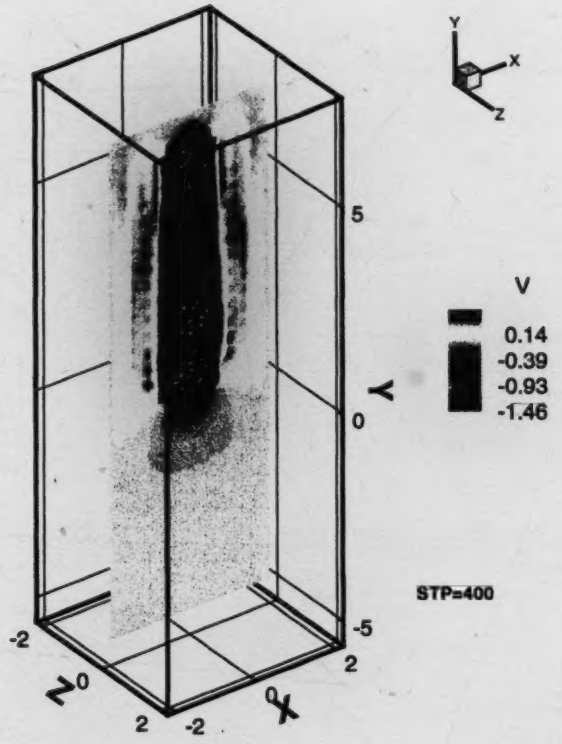

Figure 9.15: Two spheres in vertical configuration. Numerical results at t=4.0 (dimensionless).

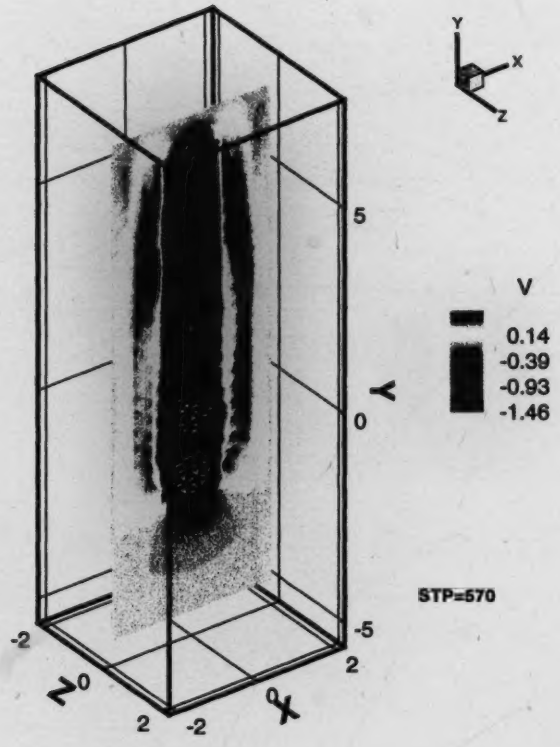

Figure 9.16: Two spheres in vertical configuration. Numerical results at t=5.7 (dimensionless).

9.3 Particle-Particle Interactions: Drafting kissing and tumbling

Particle interaction play an important role in particulate flow applications. They are a consequence of stresses and inertia. Joseph [26] described the fundamental interaction between neighboring spheres and studied their drafting, kissing and tumbling in Newtonian liquids.

To study the interaction between particles experiments were conducted in our laboratory. At a first stage direct images were taken to understand the motion of sphere kissing, drafting and tumbling. Unfortunately, we found that sometimes the two dimensional information provided by these images was not enough to determine the motion when the particles are very close together. For example, in frame 186 from Figure 9.17 we were not sure if the particles where touching and moving together, or if they were just at the same height with a distance in the depth direction. For this reason a three dimensional view was designed as explained in Chapter 7. Figures 9.17-9.22 show the 2D and 3D images obtained from our experiments with 2.2 mm steel spheres using the high speed camera.

From these experiments we can see that the particles settled under gravity. The top sphere accelerates towards the leading sphere. Then they seem to touch and both travel together for a short while. They finally separate moving apart one from the other till a stable distance is achieved. At that stable separation distance they continue settling till they reach the bottom of the tank.

Glowinski et al. [33] recently presented results obtained for the interaction of disks and spheres. They simulated the motion of two balls sedimenting in an incompressible fluid. In their simulations two spheres sediment in a rectangular channel. The computational domain was $1 \times 1 \times 4$ (dimensionless). The diameter of the two ball was 1/6 (dimensionless) and the spheres are located less than one diameter apart from each other. The mesh used to carry the simulations was quite large 2.1 10^6 vertices nodes. The collisions were handled using a repulsive force model. For their simulation the maximum Reynolds number was Re=111.46. From these simulations they present figures showing the kissing, drafting and tumbling phenomenon.

We used the results reported by Glowinski et al. [33] to compare with our simulations. Comparing with these simulations was easier than comparing with our experiments since from our experiments we could have a qualitative description of the particles motion but exact measurements were difficult to obtain. In our experiments

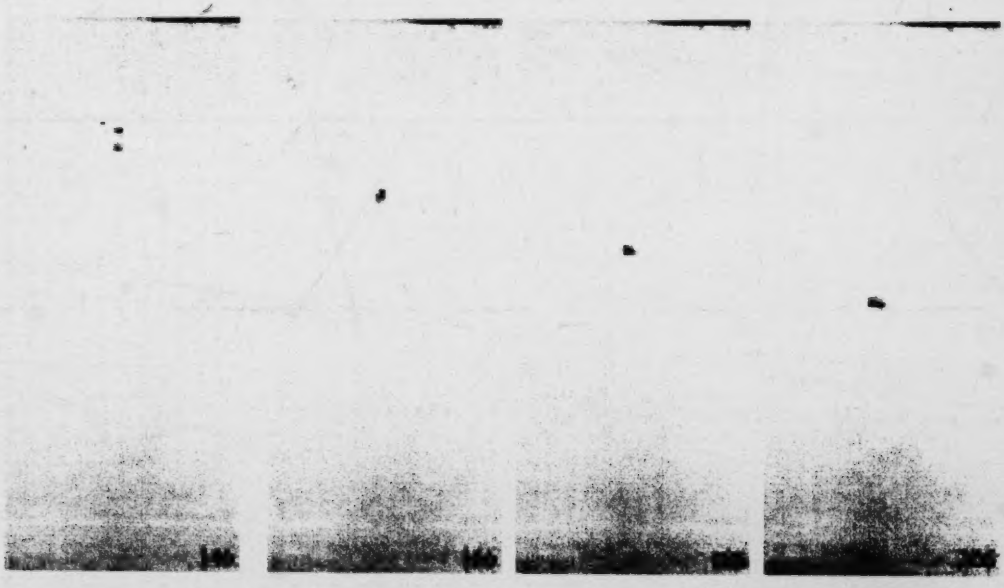

Figure 9.17: Two spheres kissing, drafting and tumbling. t=0.584s, 0.664s, 0.744 and 0.824s.

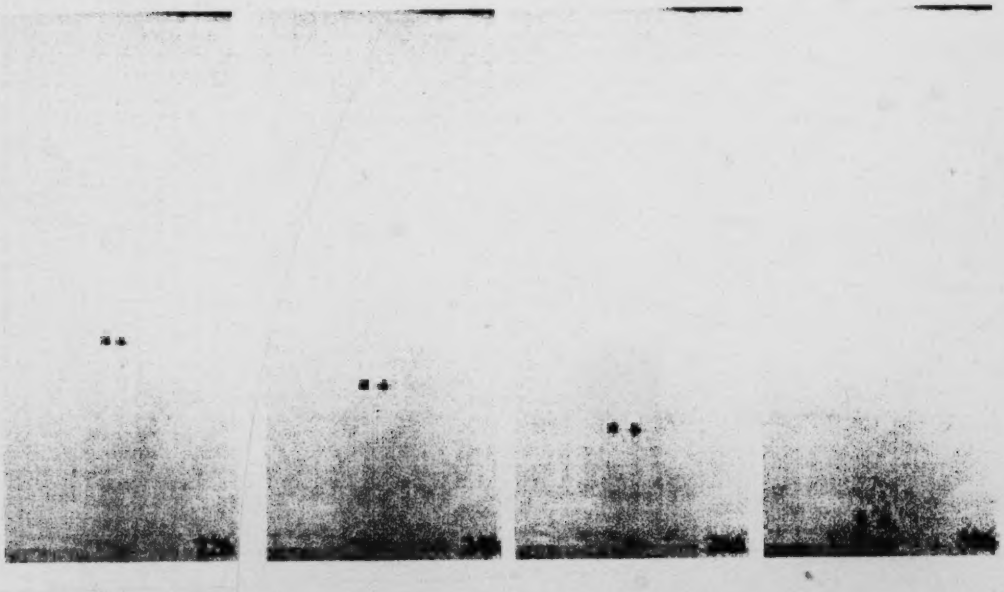

Figure 9.18: Two spheres kissing, drafting and tumbling. t=0.904, 0.984s, 1.064s and 1.224s.

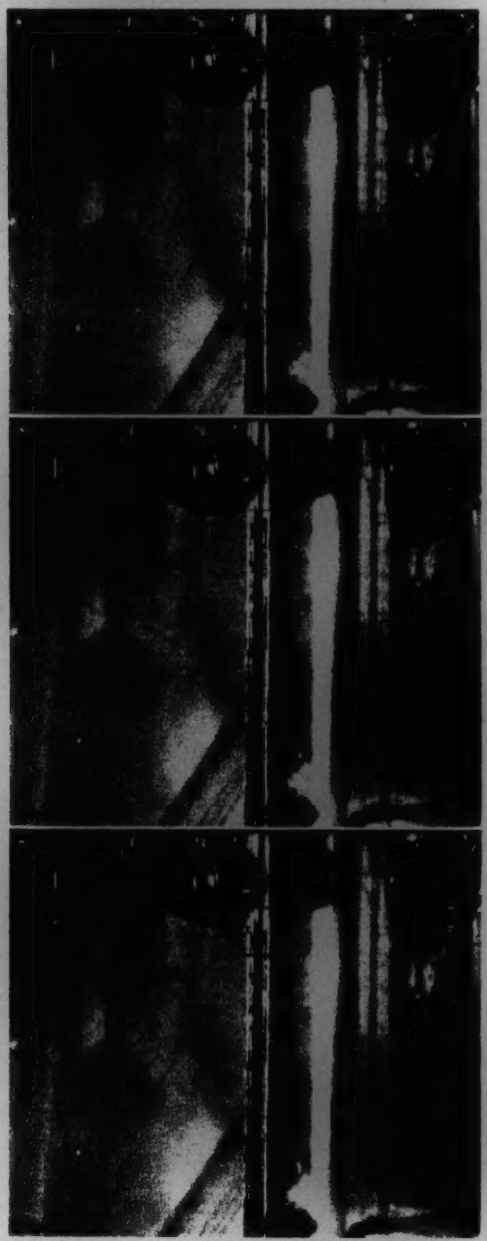

Figure 9.19: Two spheres kissing, drafting and tumbling. 3D - t=0s, 0.04s and 0.08s.

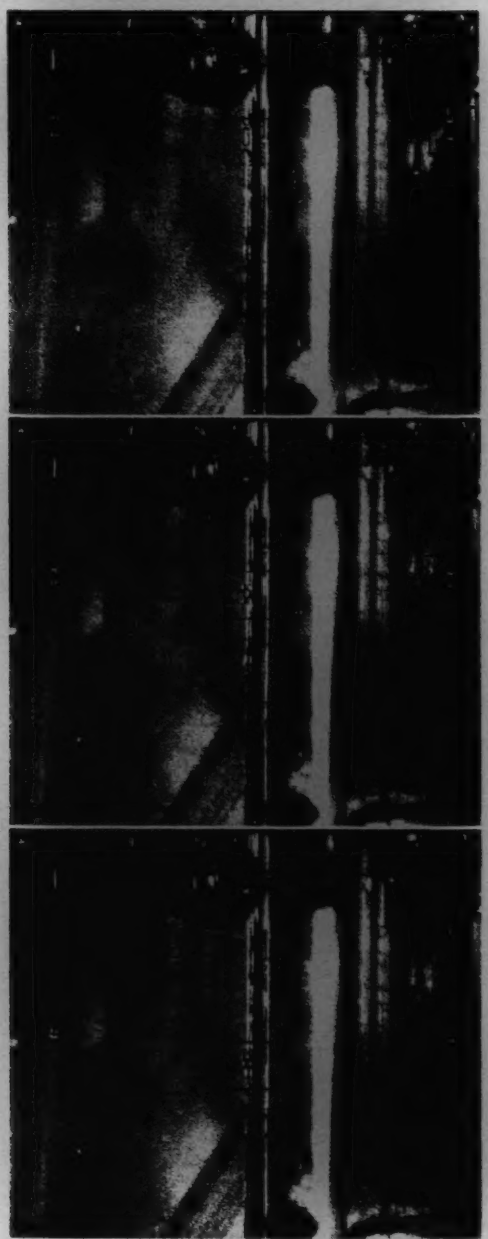

Figure 9.20: Two spheres kissing, drafting and tumbling. 3D - t=0.12s, 0.16s and 0.2s.

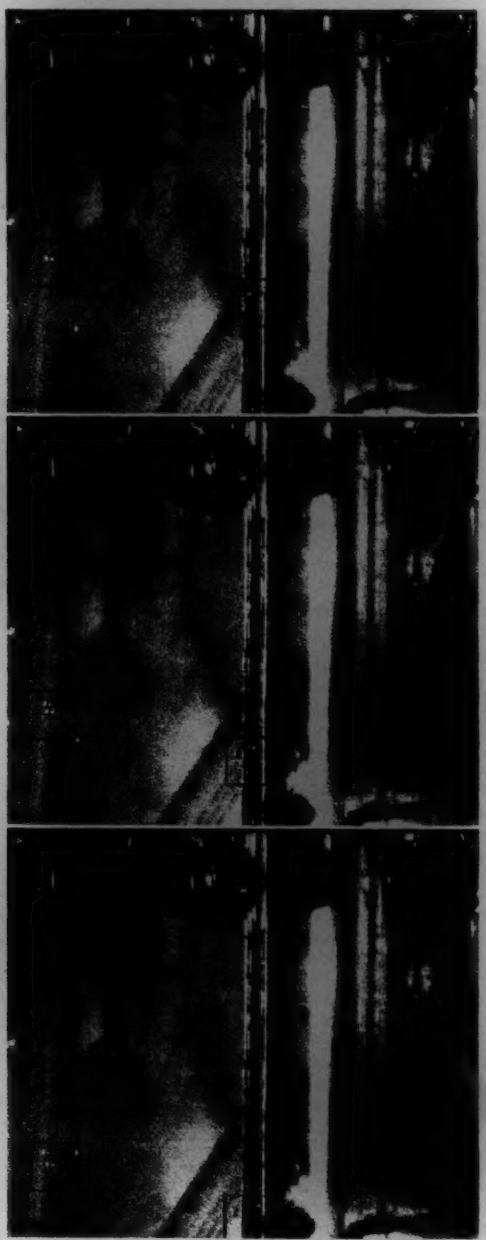

Figure 9.21: Two spheres kissing, drafting and tumbling. 3D - t=0.24s, 0.28s and 0.32s.

Figure 9.22: Two spheres kissing, drafting and tumbling. 3D - t=0.36s.

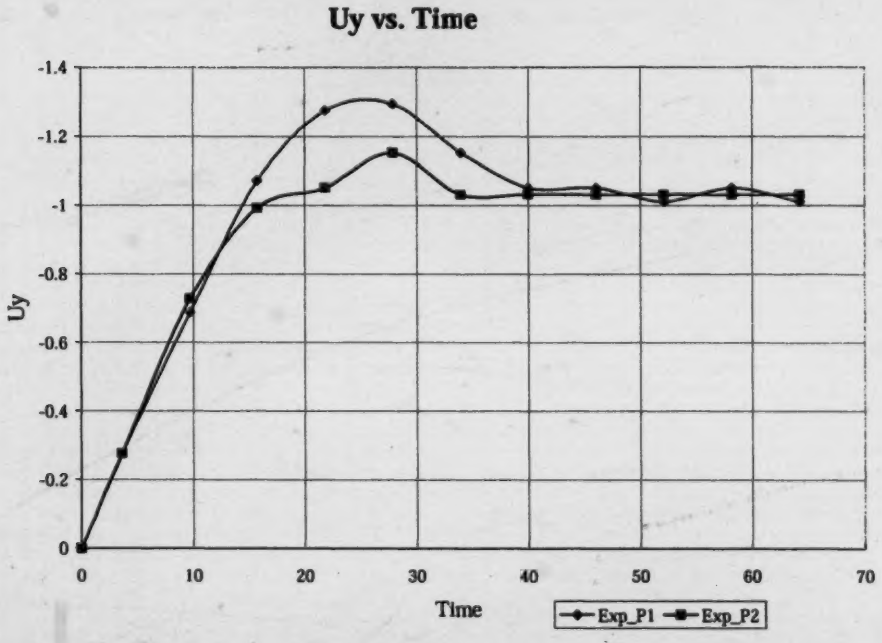

Figure 9.23: Two spheres kissing, drafting and tumbling. Experimental results. Vertical velocity vs. time (dimensionless).

one particle is released before the other and that would affect the initial motion of the second particle. Another difficulty with using the experimental results was that for such a Reynolds number, the computational mesh to accurately capture the formation of a wake behind the first sphere would have been quite large.

For this case we had to re-scale the problem since before we had made the problem dimensionless using parameters such as the Reynolds number and the Froude number, so that the terminal velocity would be scaled to the unit. Unfortunately, we could not use a mesh as refined as the one used by Glowinski since we did not have the corresponding computational resources.

To simulate the interaction between particles, simulations were conducted using a mesh of 306231 nodes and 205800 elements. The Reynolds number was set to Re=108. The density of the particles is $\rho_S = 7.80$ and density of the fluid is $\rho_F = 0.826$. The initial conditions are that the fluid and particle are at rest. No-slip conditions are prescribed on all the walls. Both spheres have radius of 0.4. Their initial positions are (x,y,z)=(0.0, 6.0, 0.0) and (0.0, 5.0,0.0) respectively.

Figures 9.24 and 9.30 show the particle's positions and velocity field obtained for this case (Re=108). As in the case of our experimental tests and of the results presented by Glowinski [33] our numerical simulations show the kissing, tumbling and drafting. The upper sphere settles slower than the bottom sphere at first. After a while the velocity of this upper sphere increases and the distances between them becomes smaller, we can see this sphere moves downwards towards the leading one. This is caused by the reduced drag the upper sphere experiments when it enters the leading sphere's wake. The two spheres then kiss and continue dropping together. This new configuration is unstable and the pair tumbles. The spheres start to separate until a stable distance is achieved. Finally the two spheres fall side by side.

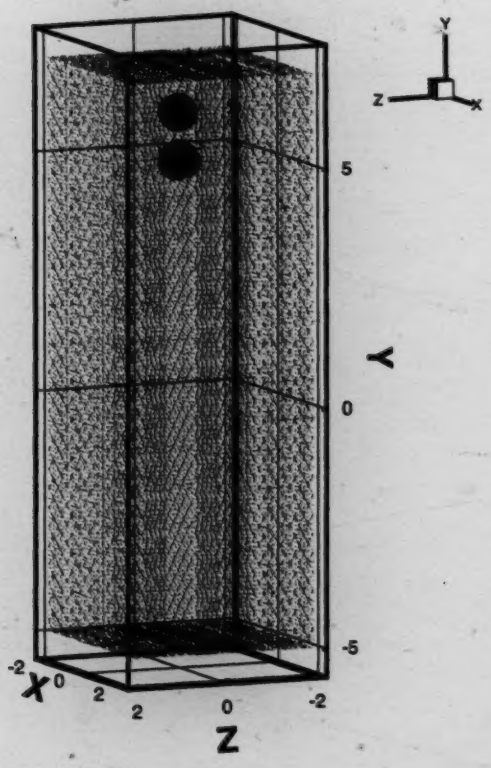

Figure 9.24: Two spheres kissing, drafting and tumbling: Numerical results for t=0.01 (dimensionless). Re=108.

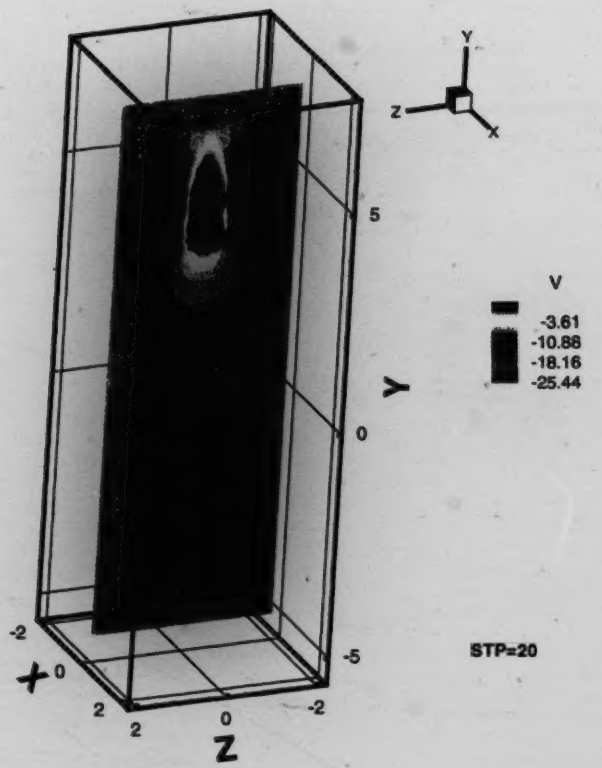

Figure 9.25: Two spheres kissing, drafting and tumbling: Numerical results for t=0.20 (dimensionless). Re=108.

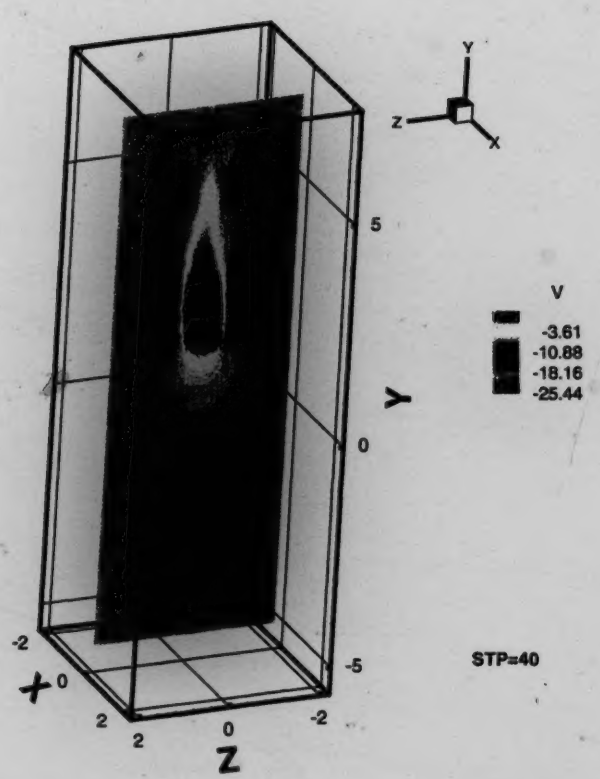

Figure 9.26: Two spheres kissing, drafting and tumbling: Numerical results for t=0.40 (dimensionless). Re=108.

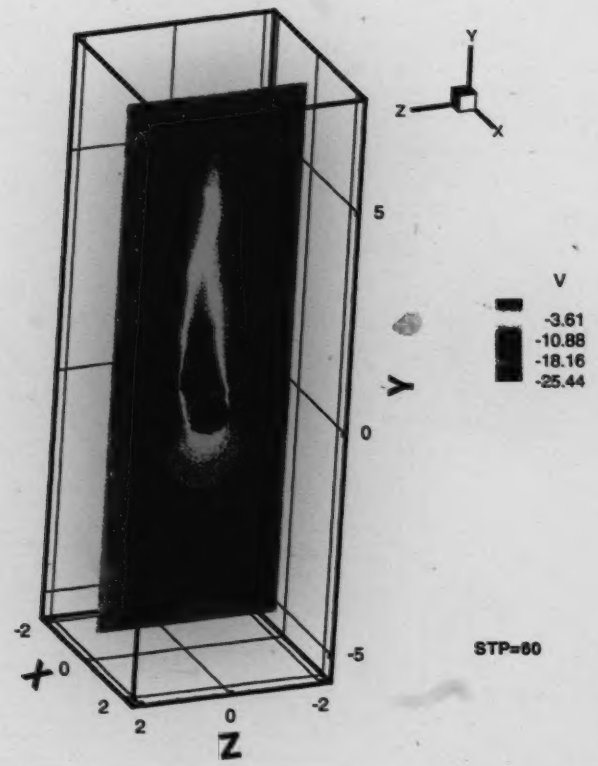

Figure 9.27: Two spheres kissing, drafting and tumbling: Numerical results for t=0.60 (dimensionless). Re=108.

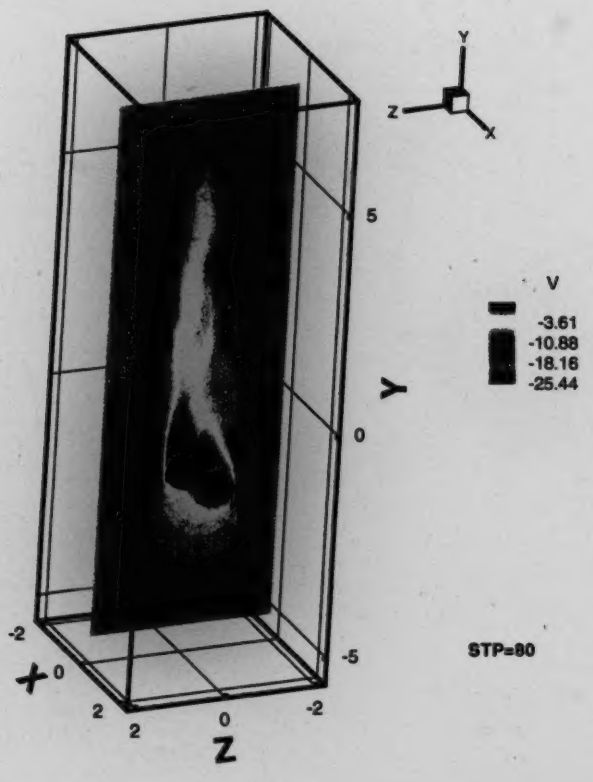

Figure 9.28: Two spheres kissing, drafting and tumbling: Numerical results for t=0.80 (dimensionless). Re=108.

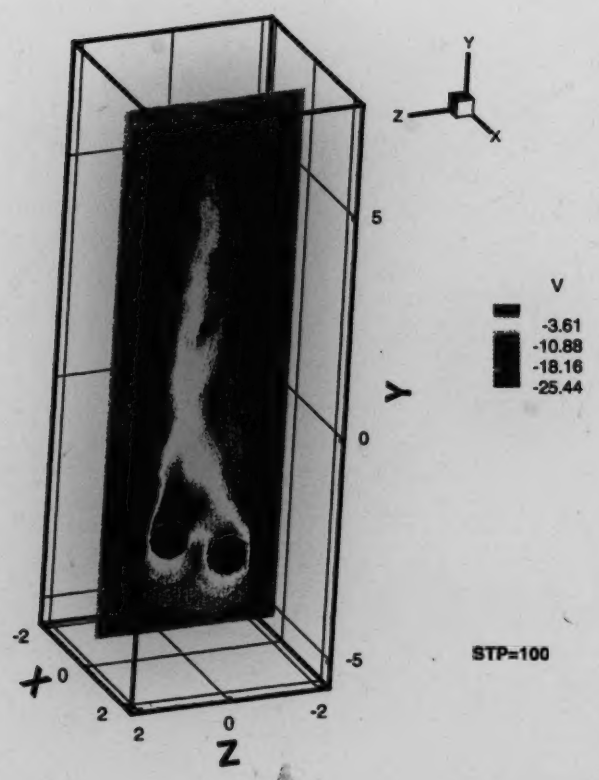

Figure 9.29: Two spheres kissing, drafting and tumbling: Numerical results for t=1.0 (dimensionless). Re=108.

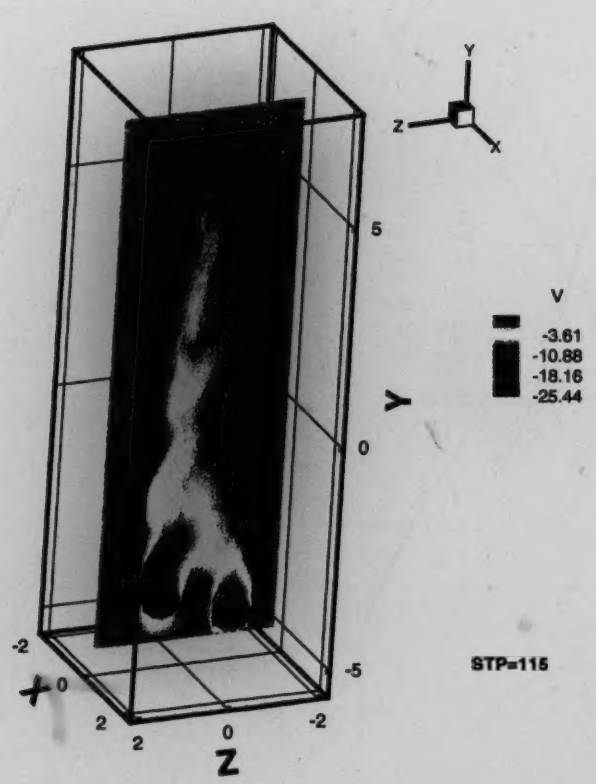

Figure 9.30: Two spheres kissing, drafting and tumbling: Numerical results for t=1.15 (dimensionless). Re=108.

Particle	X_0 (cm)	ΔX (cm) with respect to center sphere	Uy_{inf}(cm/s)
1	-0.28	-0.08	12.5
2	0	0	12.5
3	0.21	0.12	12.5

Table 9.1: Three spheres settling. Experimental setup and terminal velocity.

9.4 Three spheres in an horizontal arrangement

Experiments and numerical simulations were carried out to study the motion of three solid particles settling. In this first case the three spheres are released at the same initial vertical position and spaced horizontally with a distance of one diameter between each other. The experiments were carried out using 0.99 mm diameter steel bearings. Table 9.1 shows the initial position of each sphere relative to the center sphere, the displacement in the x component with respect to the center one (separation between center of mass of each sphere) and the terminal velocity obtained in each case. This terminal velocity is just an approximation since the value of the terminal velocity of each sphere oscillated between 12.4 and 12.7.

Figure 9.31 shows the pictures obtained with the high speed camera. Figure 9.32 showed the evolution of the terminal velocity with time. The spheres settled increasing the separation between each other till the separation distance is about 2 diameters. The three spheres reach approximately the same vertical terminal velocity.

Figures 9.33 to 9.40 show the numerical results obtained for this case. The mesh used had 306231 nodes and 205800 elements. Reynolds number was set to Re= 84.7 and Fr=1.2. These values reflect the radius used for this simulation of 0.35. The density of the solid was set to ρ_F=7800 and the density of the fluid to ρ_S=826. The timestep used was of dt=0.01. Initially the fluid is at rest and the particles are released at t=0. The Figures 9.33 and 9.37 show the particle's positions for different timesteps. Figures 9.38 to 9.40 show the evolution of the vertical velocity with time and the spheres displacement in the x and z coordinates. For this simulation the spheres move downwards with very little displacement in the other two components. We should note that the mesh used in this simulation was much smaller than the corresponding experimental tank, leaving the spheres with little space to separate even further. The sphere closer to the wall reaches a smaller terminal velocity. This could have been caused by its proximity to the wall.

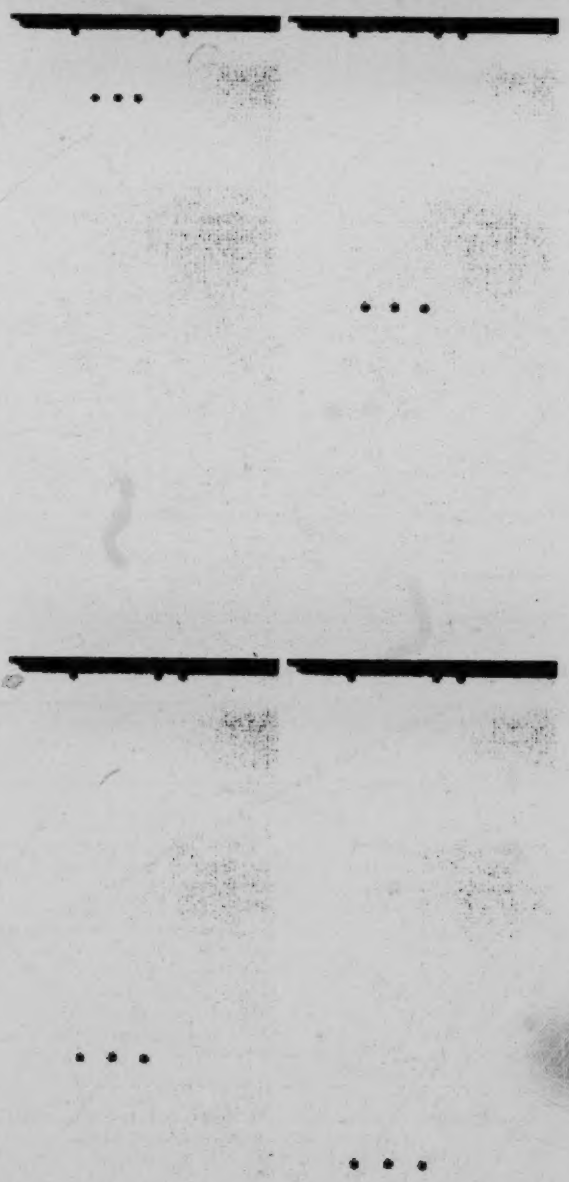

Figure 9.31: Three spheres settling. Experimental results for spheres released at the same vertical height. t=0.20s, 0.60s, 0.80s and 1.0s.

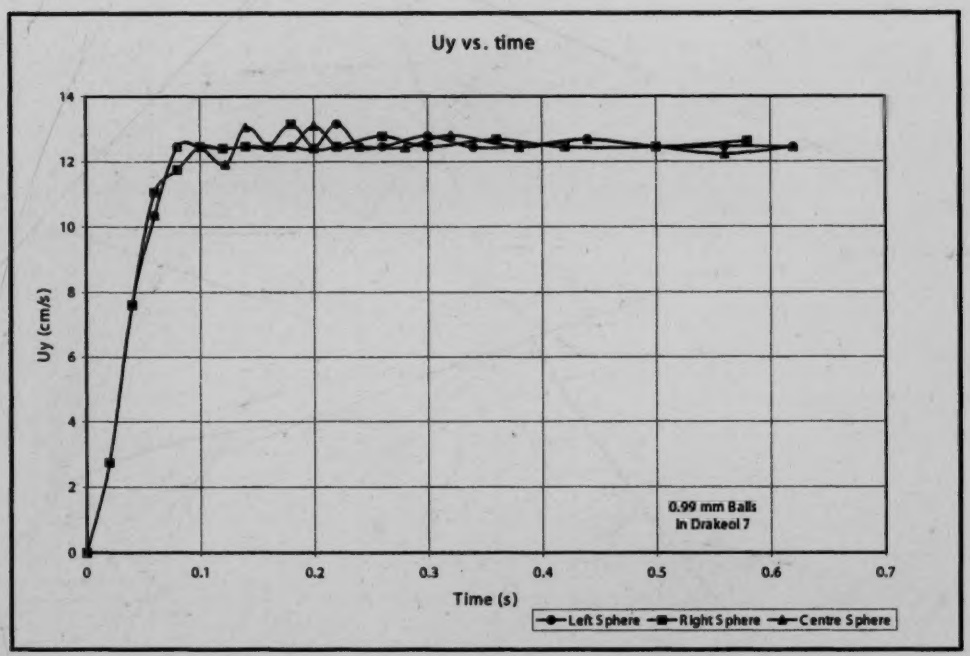

Figure 9.32: Three spheres settling. Experimental results. Terminal velocity vs. time.

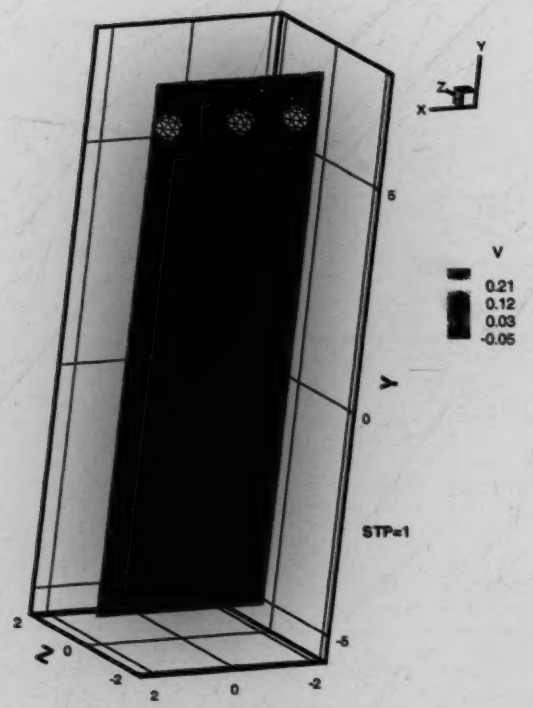

Figure 9.33: Three spheres settling. Numerical results. t=1dt (dimensionless).

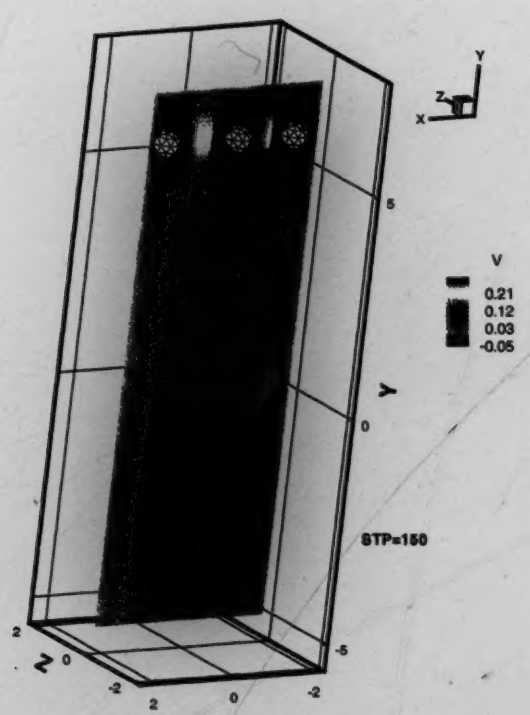

Figure 9.34: Three spheres settling. Numerical results. t=150dt (dimensionless).

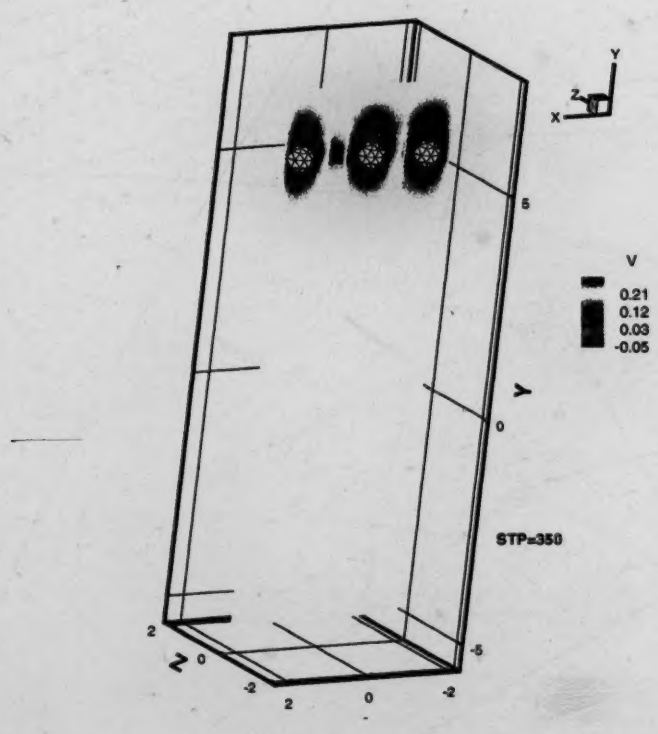

Figure 9.35: Three spheres settling. Numerical results. t=350dt (dimensionless).

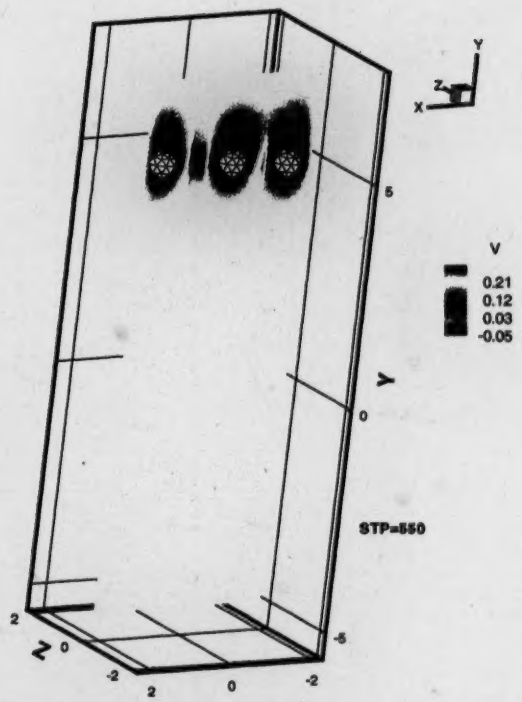

Figure 9.36: Three spheres settling. Numerical results. t=550dt (dimensionless).

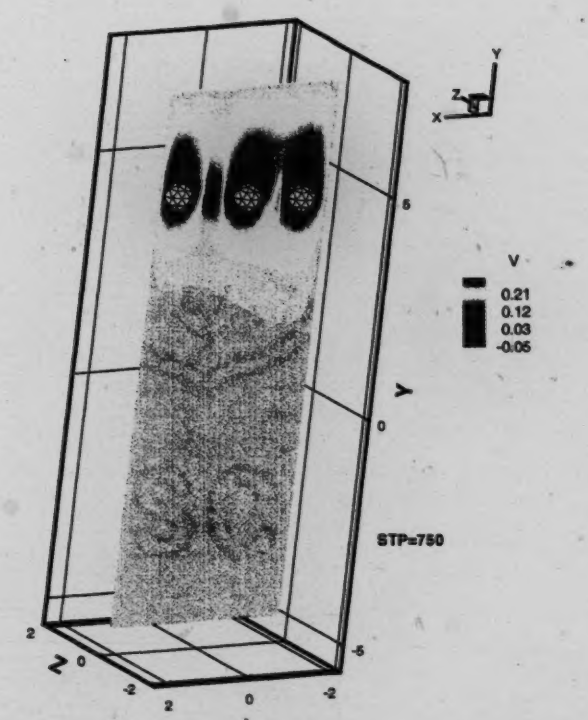

Figure 9.37: Three spheres settling. Numerical results. t=750dt (dimensionless).

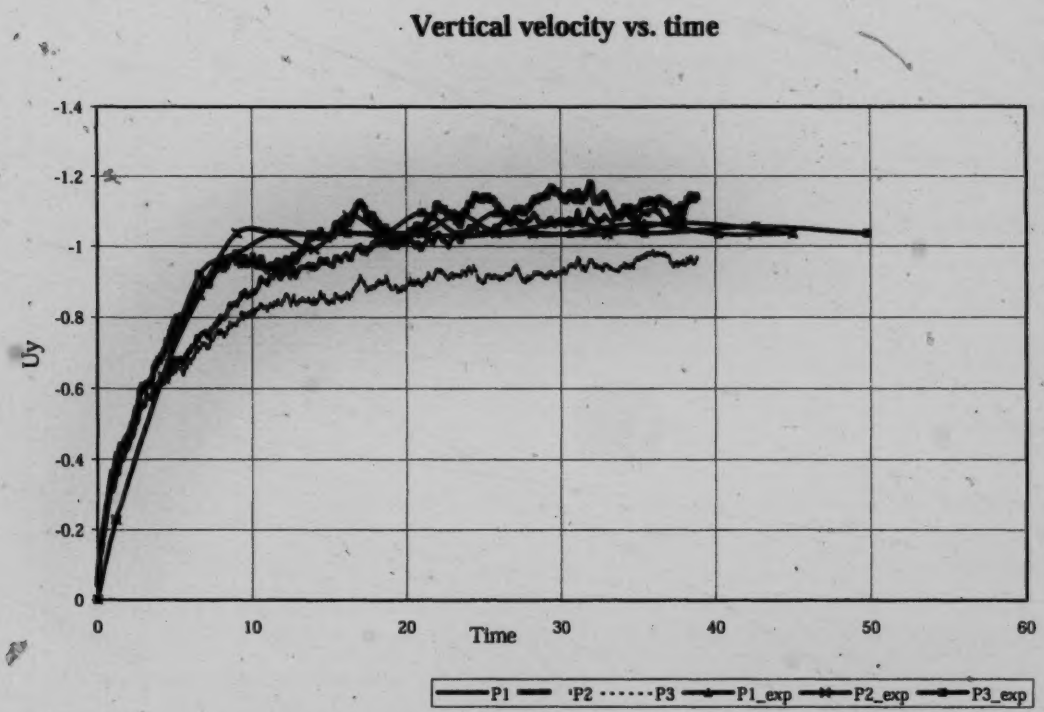

Figure 9.38: Three spheres settling. Experimental and numerical results. Terminal velocity vs. time (dimensionless).

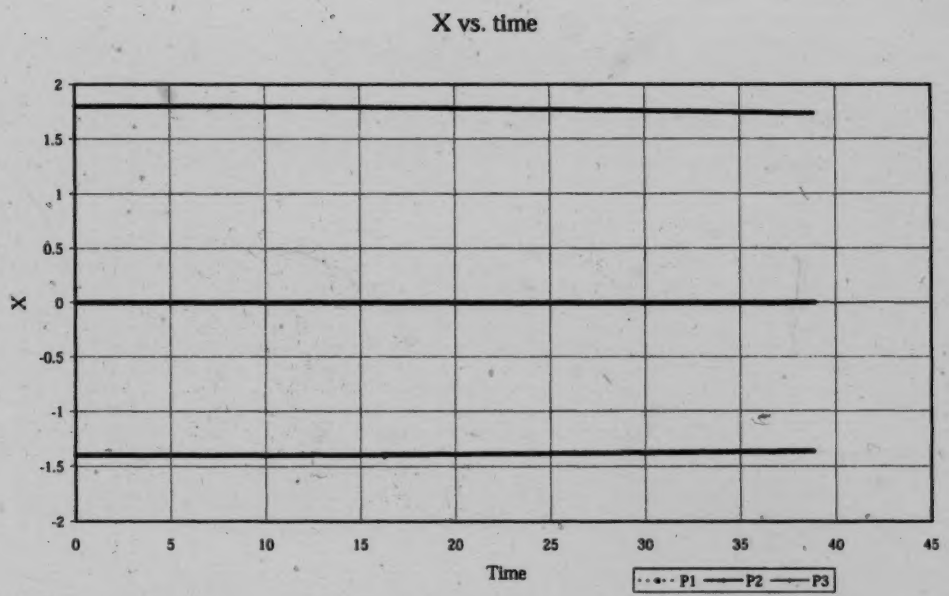

Figure 9.39: Three spheres settling. Numerical results. X displacement vs. time (dimensionless).

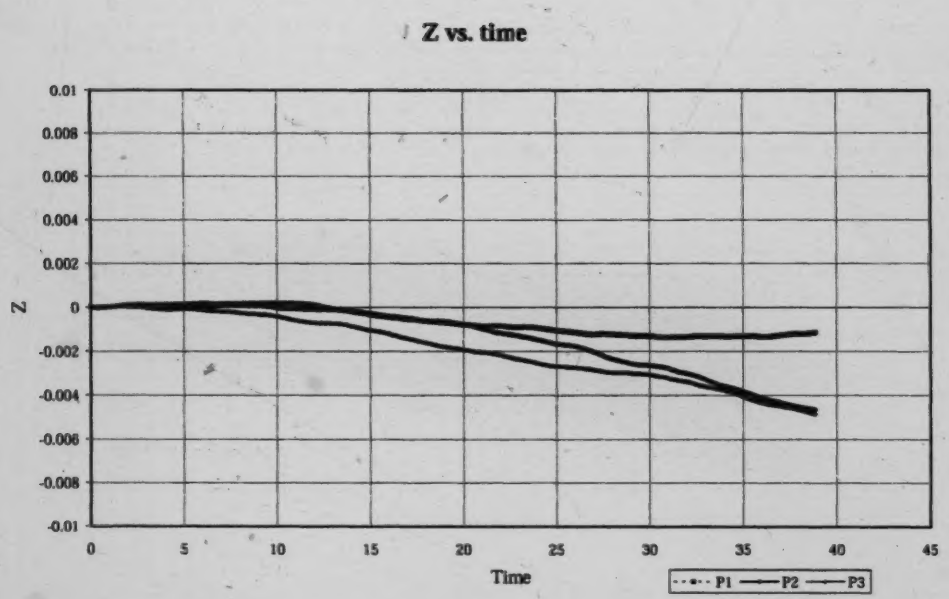

Figure 9.40: Three spheres settling. Numerical results. Z displacement vs. time (dimensionless).

9.5 Three spheres in a vertical arrangement

For this experiment three spheres are arranged vertically and released one at a time in a fluid at rest. The 3D images are presented in Figure 9.41. As soon as the spheres start dropping the second sphere approaches the leading one. They continue to drop together while the top sphere accelerates towards them. The three spheres fall together for a very short period. This configuration is unstable and the spheres tumble and separate. They spread apart and then continue to fall till they reach the bottom of the tank. To simulate this case a 306231 nodes and 205800 element mesh was used. The timestep was set to 0.01. Other input parameters such as the Reynolds and Froude numbers were: Re=110 and Fr=0.1. The ratio of densities was set to 9.44. The initial and boundary conditions were that the solids and fluid are at rest at time t=0 and the velocity was prescribed to be zero on all the walls. Figure 9.43 shows the results obtained for our simulation. Overall, the numerical simulation captures all the characteristics of the motion studied in our experiment.

9.6 27 spheres settling

Experiments were limited to a few spheres since it was extremely hard to release the sphere in a controlled fashion. Although the electromagnet technique proved to be useful it could not handle too many particles. On the other hand, published data corresponding to simulations of many particles is scarce. Commercial software widely used in the university (i.e. Fluent) could only handle one solid particle. For these reasons, the following case was not compared to any experiment. We believe its results to be correct since the goodness of the simulations had been determined for all previous cases. To simulate twenty seven spheres settling under gravity we used a mesh with 653455 nodes and 449280 elements. Reynolds was set to Re=59.35 and Froude to Fr=1.72. The ratio of density was set to 9.44. The timestep for this simulation was of 0.01. Initially the fluid and spheres are at rest. Boundary conditions are prescribed for all walls. The spheres are arranged in three groups of 9 spheres each, equally spaced and centered with respect to the numerical domain as shown in Figure 9.52. Figures 9.53 to 9.55 show the results obtained. The spheres start to drop and move horizontally in x and z direction to increase the separation between them. As they drop some collisions occurred.

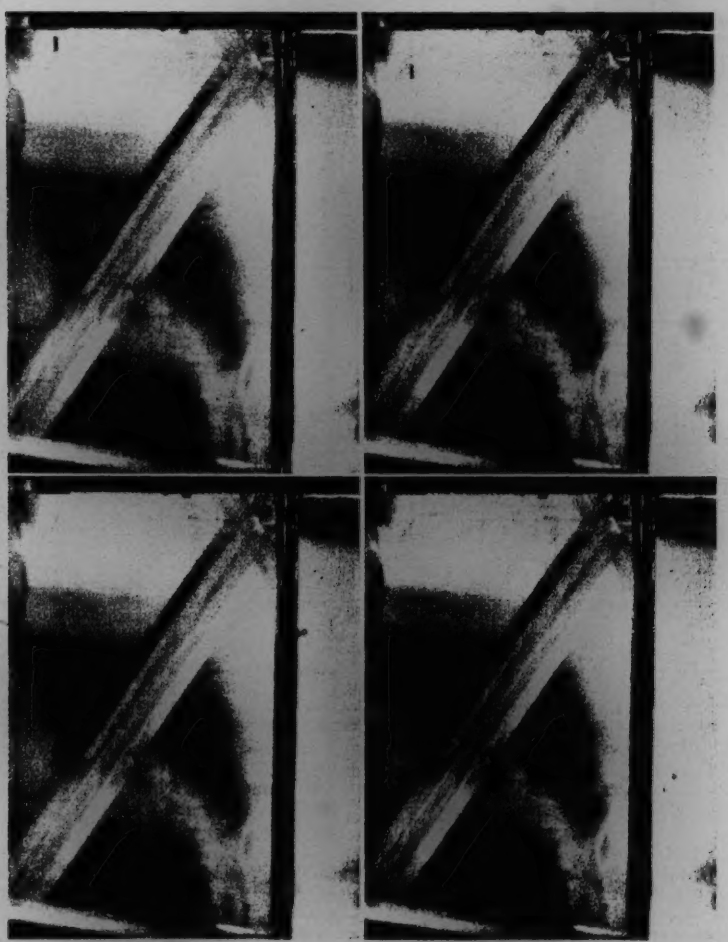

Figure 9.41: Three spheres in a vertical arrangement. Experimental results: positions for t=0.04s, 0.08s, 0.16s and 0.32s.

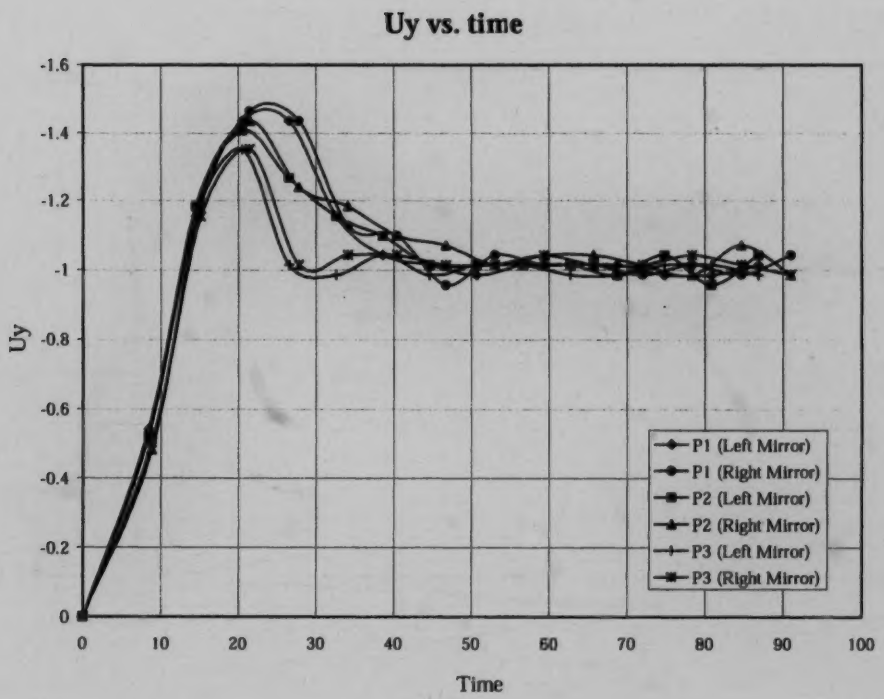

Figure 9.42: Three spheres in a vertical arrangement. Experimental results: vertical velocity vs. time.

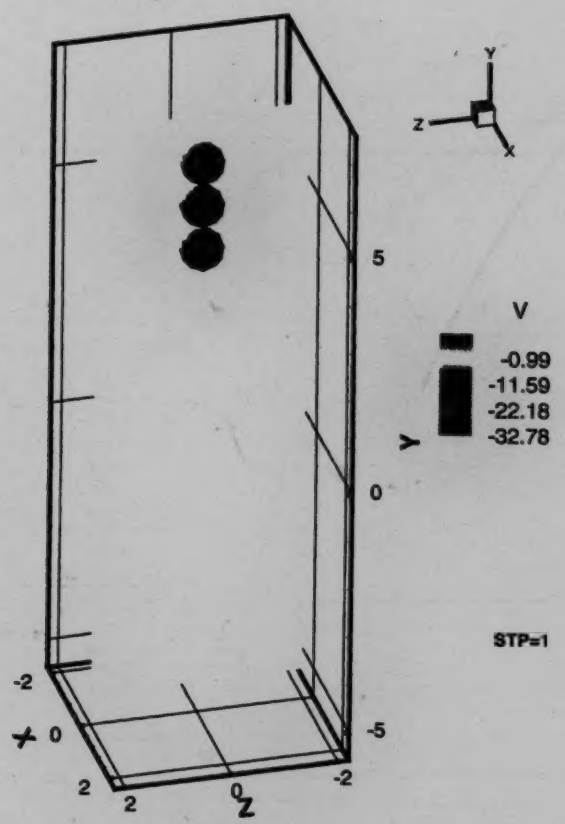

Figure 9.43: Three spheres in a vertical arrangement. Numerical results: t=1dt (dimensionless).

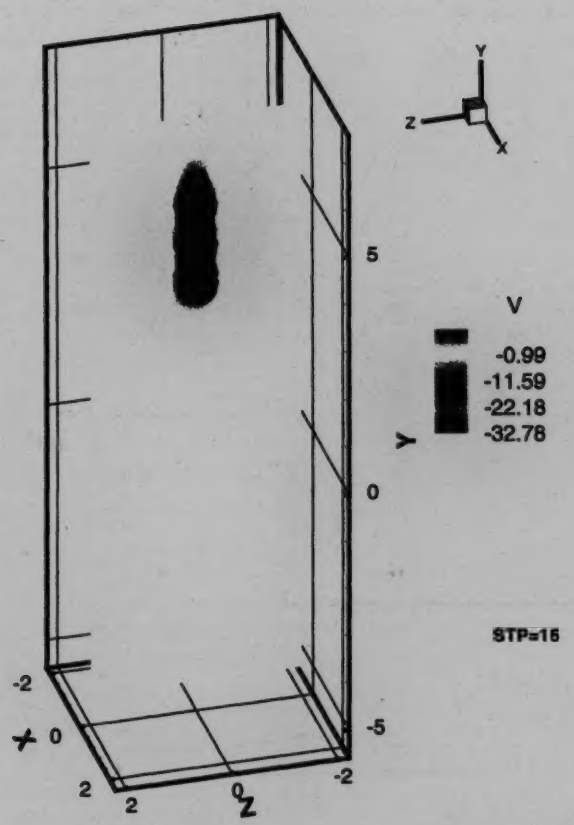

Figure 9.44: Three spheres in a vertical arrangement. Numerical results: t=15dt (dimensionless).

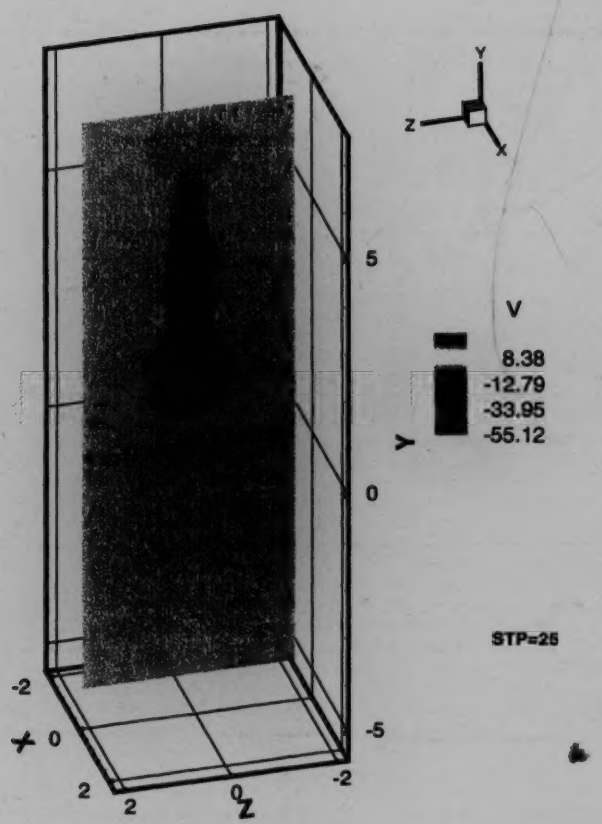

Figure 9.45: Three spheres in a vertical arrangement. Numerical results: t=25dt (dimensionless).

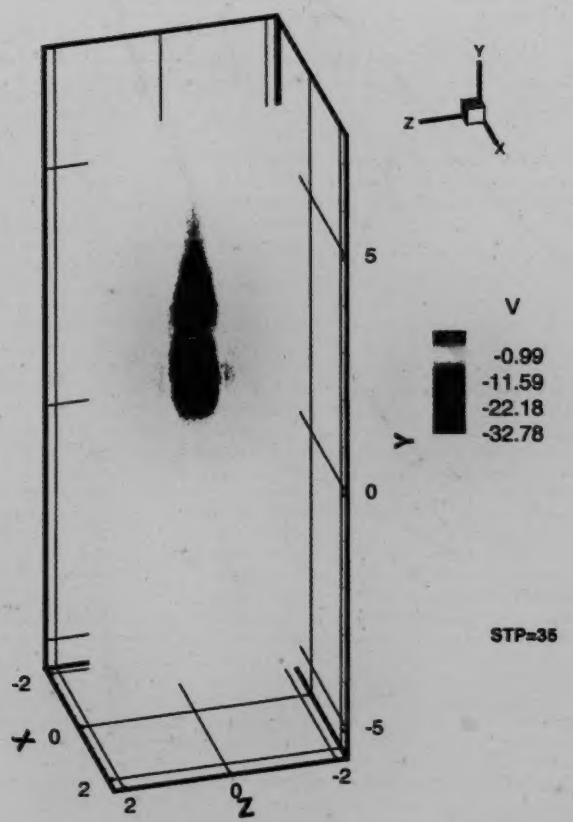

Figure 9.46: Three spheres in a vertical arrangement. Numerical results: t=35dt (dimensionless).

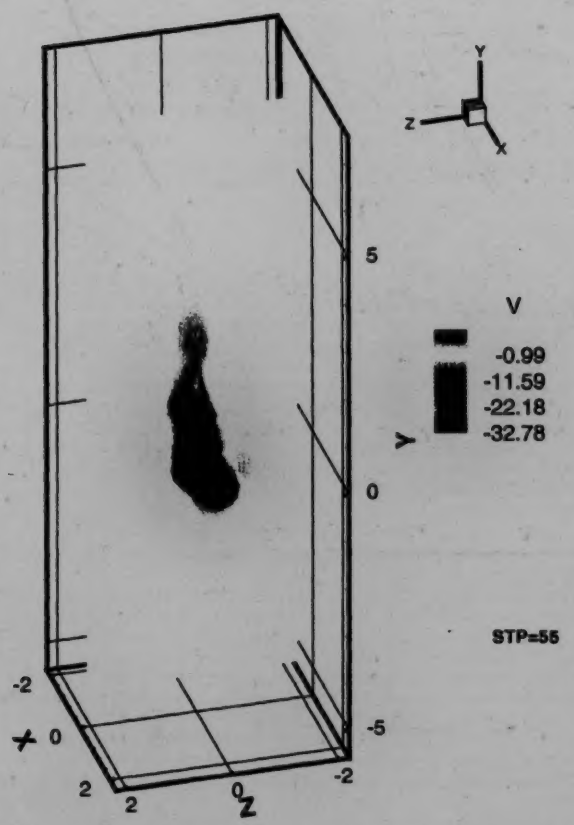

Figure 9.47: Three spheres in a vertical arrangement. Numerical results: t=55dt (dimensionless).

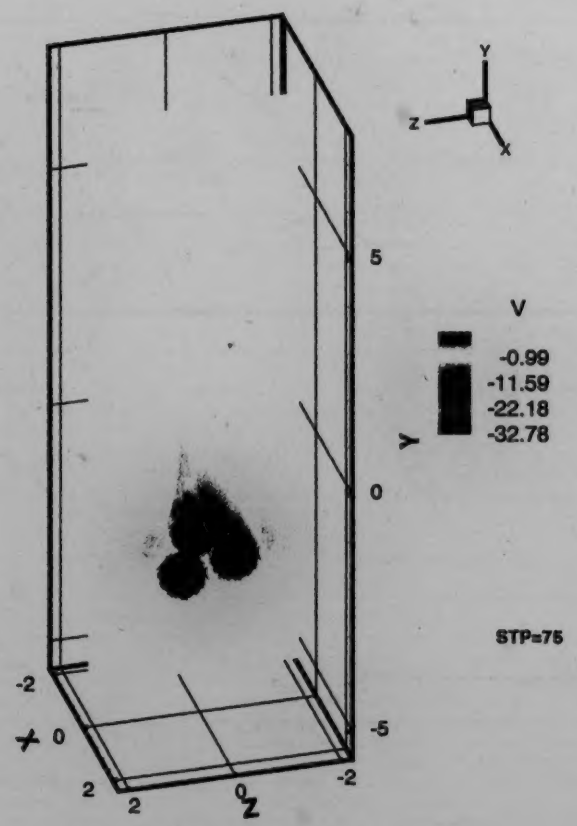

Figure 9.48: Three spheres in a vertical arrangement. Numerical results: t=75dt (dimensionless).

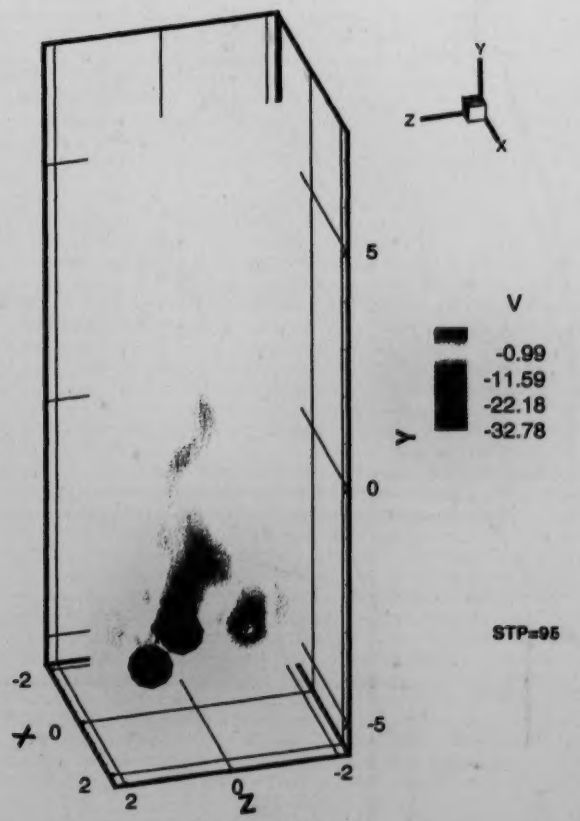

Figure 9.49: Three spheres in a vertical arrangement. Numerical results: t=95dt. Contour at X=0.12 (dimensionless).

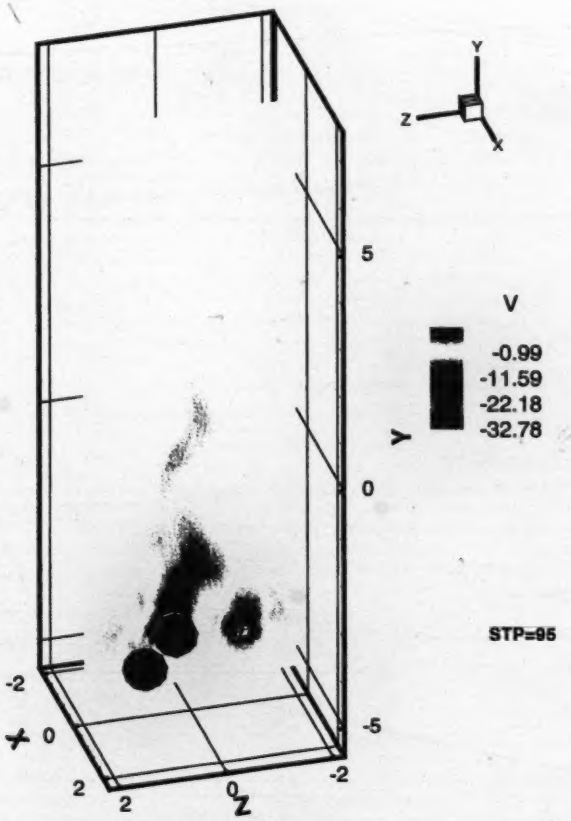

Figure 9.50: Three spheres in a vertical arrangement. Numerical results: t=95dt (dimensionless). Side view.

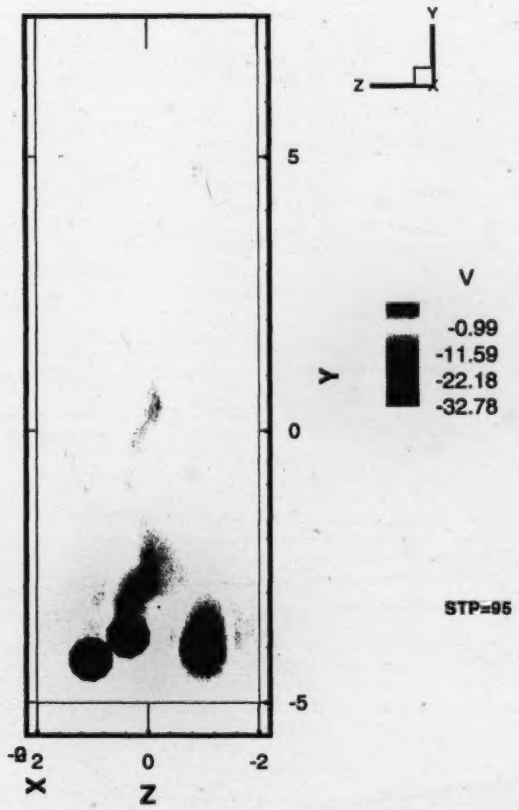

Figure 9.51: Three spheres in a vertical arrangement. Numerical results: t=95dt. Contour at X=-0.10 (dimensionless).

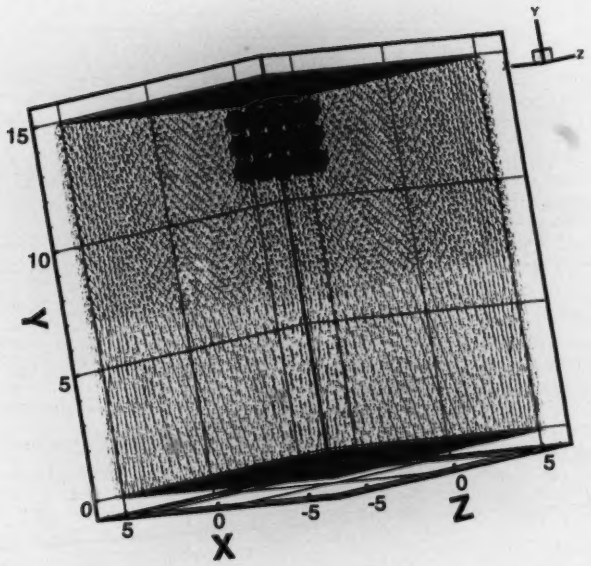

Figure 9.52: Settling of 27 spheres. Computational domain.

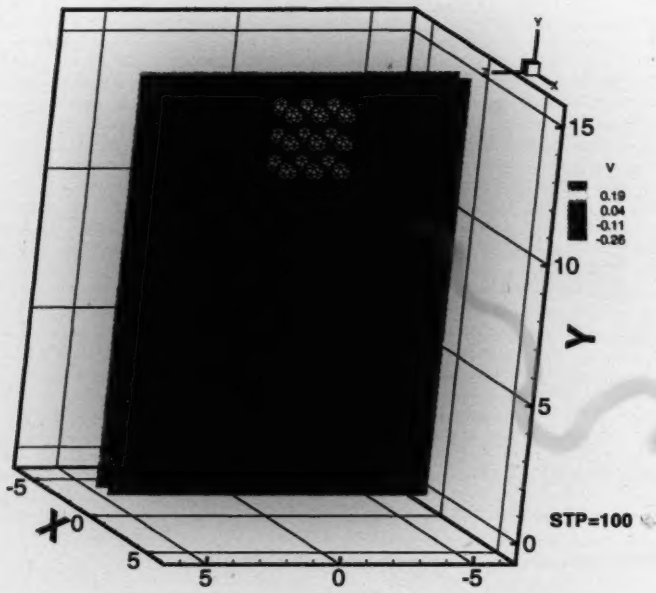

Figure 9.53: Settling of 27 spheres. t=1.00 (dimensionless).

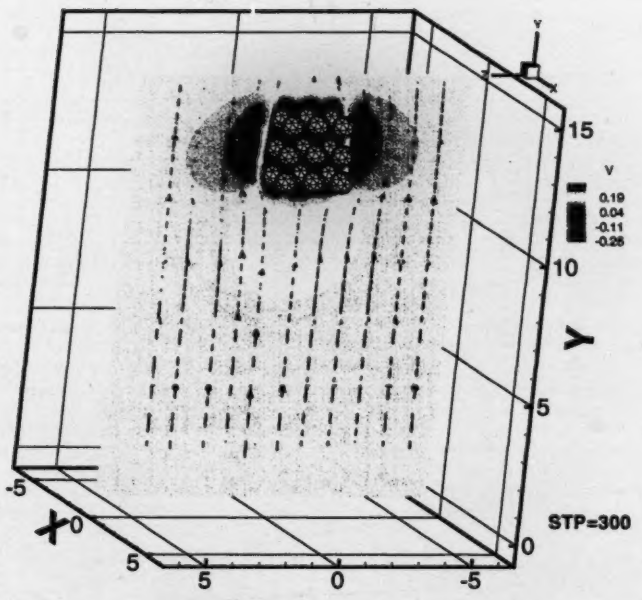

Figure 9.54: Settling of 27 spheres. t=3.00 (dimensionless).

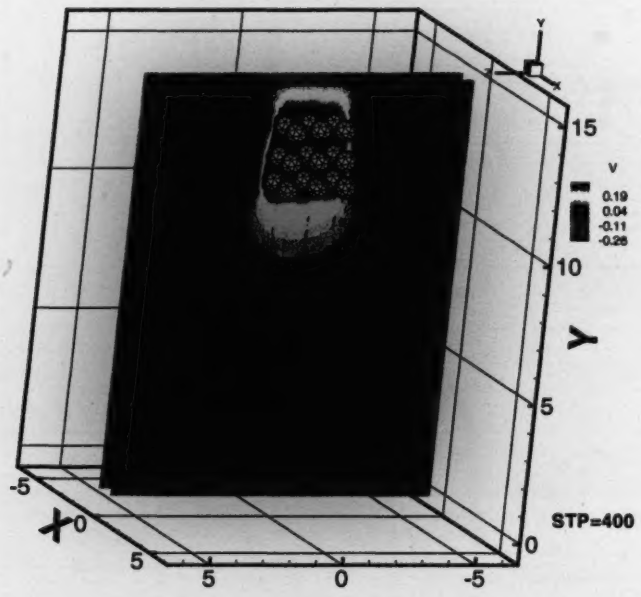

Figure 9.55: Settling of 27 spheres. t=4.00 (dimensionless).

9.7 Performance of the parallel code

Table 9.2 shows the average time spent per pass of the conjugate gradient (with an average of 7 iterations to achieve convergence) and the speedup factor for different number of processors used in each run. For our comparison the speedup factor $S(n)$, is calculated as:

$$S(n) = \frac{time_{serial}}{time_{parallel}} \tag{9.1}$$

The test case used to compare execution times was that of one particle sedimenting under the influence of gravity. The mesh had 179401 nodes and 120000 elements. The cases were run on an SGI Origin model 2000 with 44 Processors (195MHZ), 12 Gbytes of main memory.

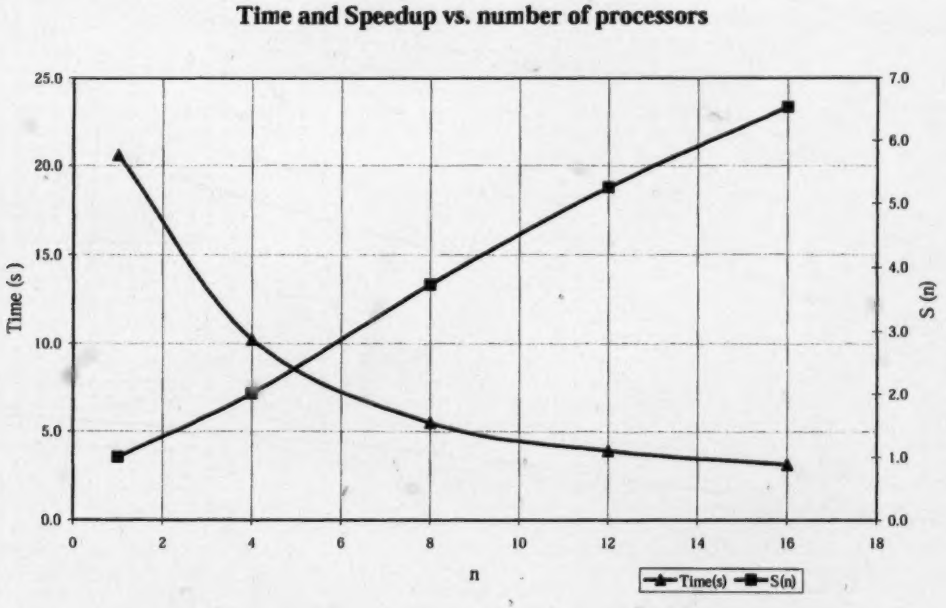

Figure 9.56: Time and Speedup vs. number of processors

Table 9.3 shows the good scalability properties of the algorithm. For larger problems the total CPU time spent in the conjugate gradient increases and so does the speedup factor.

Num. of Processors n	Average Time (s)	Speedup Factor $S(n)$
1	20.60	1.00
4	10.25	2.01
8	5.51	3.74
12	3.92	5.26
16	3.15	6.54

Table 9.2: Number of processors, average time per pass of the conjugate gradient and speedup

Mesh size	%Time of CG	$S(n)$ for 16 processors
32579	67.6	6.0
179401	83.5	6.54
396223	90.1	7.09

Table 9.3: Comparison of mesh size (number of nodes) with time spent in CG and speedups achieved

Chapter 10

Conclusions

A new algorithm to solve for the equations that describe particulate flow is presented. The method that we suggested in the present paper is a fictitious domain Eulerian method for direct simulation of particulate flows. The main advantages in comparison to other methods of the same type (in particular the method that we used as a starting point for the present study, proposed by Glowinski et al., [33]) are that:

1. It defines a global Lagrange multiplier, $\boldsymbol{\lambda}$, whose physical meaning is of an additional velocity field that imposes the rigid body motion. This allows us to completely eliminate the Lagrangian grids used by [33] to discretize the distributed Lagrange multipliers in their case. This allows for the use of basically only one solver on relatively regular grids which facilitates the parallel implementation of the method.

2. The global Lagrange multiplier, together with a proper choice for the inner product that is used to impose the rigid body constraint allows to avoid the need to solve the linear system for computing the distributed Lagrange multipliers in each particle. Although this system is not large, it eventually is to be solved on each iteration. If the flow contains many particles, the savings can be significant.

3. The present algorithm employs a second order in space, incremental projection scheme for the resolution of the generalized Stokes problem which, as our numerical experience shows, performs better than a first order scheme. The spatial discretization is also second order accurate for the velocity and first order accurate for the pressure.

In addition to these advantages, this method retains all the advantages of the method proposed by Glowinski et al. [33].

The new algorithm was first used to implement a program to solve for one solid particle. This program was then extended to handle many solid particles. The mathematical formulation remained the same but a collision strategy had to be put into place to avoid particles from interpenetrating each other. Finally, the code was parallelized and its performance was studied for different size problems and its speedup measured. In all cases the program was validated against experimental data obtained in the laboratory or from published literature.

Future work could involve improving the collision algorithm using local mesh refinement to provide a finer zone to detect and handle collisions. Thus, we would have a multi-scale problem and a more physical representation of those collisions could be put into place. The conjugate gradient preconditioner could be enhanced. If the problems were restricted to use uniform grids then domain decomposition techniques could be used for the parallel solver. A variety of physical settings to better understand and visualize particulate flow can now be simulated.

Bibliography

[1] Achenbach, E. Vortex shedding from spheres. *J. Fluid Mech. 62* (1974), 209–221.

[2] Allievi, A., and Bermejo, R. Finite element modified method of characteristics for the Navier-Stokes equations. *J. Num. Meth. Fluids 32* (2000), 439–464.

[3] Aris, R. *Vectors, Tensors and the Basic Equations of Fluid Mechanics.* Dover Publications, 1989.

[4] Armaly, B., Durst, F., Pereira, J., and Schonı ,, B. Experimental and theoretical investigation of backward-facing step flow. *J. Fluid Mech. 127* (1983), 473–496.

[5] Astrakhatsev, G. Methods of fictitious domains for a second-order elliptic equation with natural boundary conditions. *USSR Computational Math. and Math. Physics 18* (1978), 114–121.

[6] Babuska, I. Error-bound for finite element method. *Numer. Math. 16* (1971), 322–333.

[7] Barnocky, Guy, and Davis, Robert. The influence of pressure-dependent density and viscosity on the elastohydrodynamic collision and rebound of two sphere. *J. Fluid Mech. 209* (1989), 501–519.

[8] Boukir, K., Maday, Y., Metivet, B., and Razafindrakoto, E. A high order characteristic/finite element method for the incompressible Navier-Stokes equations. *Int. J. Num. Meth. Fluids 25* (1997), 1421–1454.

[9] Brezzi, F. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *R.A.I.R.O 8* (1974), 129–151.

[10] Bubzee, B., Dorr, F., George, J., and Golub, G. The direct solution of the discrete Poison equation on irregular regions. *SIAM J. Num. Anal. 8* (1971), 722–736.

[11] Buscaglia, G.C., and Dari, E.A. Implementation of the Lagrange-Galerkin method for the incompressible Navier-Stokes equations. *Int. J. for Num. Meth. Fluids 15* (1992), 23–36.

[12] Canuto, C., and Quateroni, A. *Spectral methods in fluid dynamics.* Springer-Verlag, New York, 1988.

[13] Catlett, C. The Teragrid: A premier. Tech. rep., Teragrid, National Science Foundation., 2002.

[14] Celia, M., Russell, T., Herrera, I., and Ewing, R. An Eularian-Lagrange adjoing method for the advection-diffusion equation. *Adv. Water Resourses 13* (1990), 187–206.

[15] Chen, T., Minev, P.D., and Nandakumar, K. A finite element technique for multifluid incompressible flow using eulerian grids. submitted to *J. Comp. Phys.*

[16] Chorin, A. Numerical simulation of the Navier-Stokes equations. *Math. Comput. 22* (1968), 745–762.

[17] Clift, R., Grace, J., and Weber, M. *Bubbles, Drops and Particles.* Academic Press, 1978.

[18] Cuvelier, C., Segal, A., and van Steenhoven, A. *Finite Element Methods and Navier Stokes Equations.* D. Reidel Publishing Co., Dordrecht, 1986.

[19] Davie, C. Computational modelling of well bore phenomena. Master's thesis, Department of Civil Engineering, University of Glasgow, 1999.

[20] Davis, R., and Serayssol, J. The elastohydrodynamic collision and rebound of two sphere. *J. Fluid Mech. 163* (1986), 479–497.

[21] de Vosse, F.van, and Minev, P. Spectral element methods: theory and applications. *Int. J. Numer. Meth. Fluids 22* (1996), 673–688.

[22] Dennis, S., and Walker, J. Calculation of the steady flow past a sphere at low and moderate reynolds numbers. *J. Fluid Mech. 48* (1971), 771–789.

[23] Diaz-Goano, C., Minev, P., and Nandakumar, K. A Lagrange multipliers/fictitious domain approach to particulate flows. In *Lecture Notes in Computer Science* (Sozopol, Bulgaria, June 2001), Wasniewski Margenov and Yalamov, Eds., Third Intern. Conf. on Large-Scale Scientific Computing, Springer, p. 409. Vol. 2179.

[24] Donea, J., Giulani, S., Laval, H., and Quartapelle, L. Finite element solution of unsteady Navier-Stokes equations by a fractional step method. *Comp. Meht. Appl. Mech. Eng. 30* (1982), 53–73.

[25] Dongarra, J., Duff, I., Sorensen, D., and Van der Vorst, H. *Numerical Linear Algebra for High-Performance Computers.* SIAM, 1998.

[26] Feng, J., Hu, H., and Joseph, D. Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part I: Sedimentation. *Journal of Fluid Mechanics 261* (1994), 95–134.

[27] Fortin, M., and Glowinski, R. *The Augmented Lagrangian Method.* North-Holland, Amsterdam, 1983.

[28] Fosdick, L., and Jessup, E. An overview of scientific computing. *High Performance Scientific Computing - University of Colorado 31* (1995), 1–29.

[29] Ghidersa, B., and Dusek, J. Breaking the axisymmetry and onset of unsteadiness in the wake of a sphere. *J. Fluid Mech. 423* (2000), 33–69.

[30] Girault, V., and Raviart, P. *Finite element methods for Navier-Stokes equations.* Springer-Verlag, New York, 1986.

[31] Glowinski, R., Kuznetsov, Y., Rossi, T., and Toivanen, J. A fictitious domain method with distributed Lagrange multipliers. P. Tarvainen P. Neittaanmäki, T. Tiihonen, Ed., Third European Conference on Numerical Mathematics and Advanced Applications, World Scientific, pp. 733–742. Vol. 2179.

[32] Glowinski, R., Pan, T., Hesla, T., and Joseph, D. A distributed Lagrange multiplier/fictitious domain method for particulate flows. *Int. J. Multiphase Flow 25* (2000), 755–794.

[33] Glowinski, R., Pan, T., Hesla, T., Joseph, D., and Periaux, J. A fictitious domain approach to the direct numerical simulation of incompressible viscous

flow past moving rigid bodies: Application to particulate flow. *J. Comput. Phys 169* (2001), 363–426.

[34] Glowinski, R., Pan, T., and J.Periaux. A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering 112* (1994), 133–148.

[35] Goldman, A., Cox, R., and Brenner, H. Slow viscous motion of a sphere parellel to a plane wall. i. motion through quiescent fluid. *Chem. Eng. Sci. 22* (1967), 637–651.

[36] Gresho, P., and Chan, S. On the theory of semi-implicit projection methods for voscous incopressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory. *J. Num. Meth. Fluids 11* (1990), 587–620.

[37] Gresho, P., and Chan, S. On the theory of semi-implicit projection methods for voscous incopressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation. *J. Num. Meth. Fluids 11* (1990), 621–659.

[38] Guermond, J.L., and Minev, P.D. Analysis of a projection/characteristic scheme for incompressible flow. *J. Comp. Phys.* To appear.

[39] Hu, H., Joseph, D., and Crochet, J. Direct simulation of fluid particle motions. *Theor.Comput. Fluid Dynamics 3* (1992), 285.

[40] Hu, H., Patankar, N., and Zhu, M. Direct numerical simulation of fluid-solid systems using the arbitrary Lagrangian-Eulerian technique. *Journal of Computational Physics 169* (2001), 427–462.

[41] Hu, H. H. Direct simulation of flows of solid-liquid mixtures. *Int. J. Multiphase Flow 22* (1996), 335–352.

[42] Isaacson, E., and Keller, H. Bishop. *Analysis of numerical methods.* John Willey and Sons, New York, 1966.

[43] Iwatsu, R., Hyun, J., and Kuwahara, K. Analysis of three-dimensional flow calculations in a driven cavity. *Fluid Dynamics Research 6* (1990), 91–102.

[44] Johnson, T. A., and Patel, V.C. Flow past a sphere up to Reynolds number of 300. *J. Fluid Mech. 378* (1999), 19–70.

[45] Johnson, T. A., and Tezduyar, T.E. 3D simulation of fluid-particle interactions with the number of particles reaching 100. *Comp. Meth. Appl. Mech. Eng. 145* (1997), 301.

[46] Karniadakis, G., Israeli, M., and Orszag, S. High-order splitting shemes for the incompressible navier-stokes equations. *J. Comp. Phys. 97* (1991), 414–443.

[47] Kim, H., and Durbin, P. Observation of the frequency in a sphere wake and of drag increase by acoustic excitation. *Phys. Fluids 31* (1988), 3260–3264.

[48] Kim, J., and Moin, P. Application of fractional-step method to incompressible Navier-Stokes equations. *J. Comp. Phys. 59* (1985), 308–323.

[49] Laval, H., and l. Quartapelle. A fractional-step Taylor-Garlekin method for unseady incompressible flows. *Int. J. Numer. Meth. Fluids 11* (1990), 501–513.

[50] Leveque, R. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J.Num. Anal. 31* (1994), 1019–1044.

[51] Maday, Y., Patera, A., and Ronquist, E. An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow. *J. Sci. Comp. 5* (1990), 263–292.

[52] Masliyah, J., and Epstein, N. Numerical solution of a uniform flow over a sphere at intermediates Reynolds numbers. *Phys. Fluids 14* (1971), 750–751.

[53] Minev, L. Timmermans P., and van de Vosse, F. An approximate projection scheme for incompressible flow using spectral elements. *Int. J. for Num. Meth. Fluids 22* (1996), 673–688.

[54] Minev, P., and Ethier, C. A characteristic/finite element algorithm for the 3-D Navier-Stokes equations using unstructured grids. *Comp. Meth. Appl. Mech. Engng. 178* (1999), 39–50.

[55] Minev, P., Vosse, F. Van De, Timmermans, L., and Steenhoven, A. Van. A splitting algorithm for thermally-driven flow problems. *Int. J. Num. Meth. Heat and Fluid Flow 6* (1996), 51–60.

[56] Minev, P.D., and Ethier, C.R. A semi-implicit projection algorithm for the Navier-Stokes equations with application to flows in complex geometries. In *Notes on Numerical Fluid Mechanics* (Braunschweig/Wiesbaden, 1999), M. Griebel *et al.*, Ed., Vieweg, pp. 223–231. 73.

[57] Mordant, N., and Pinton, J. Velocity measurement of a settling sphere. *The European Physical Journal B. 18* (2000), 343–352.

[58] Nakamura, Isao. Steady wake behind a sphere. *Phys. Fluids 19* (1975), 5–9.

[59] Natarajan, R., and Acrivos, A. The instability of steady flow past spheres and disks. *J. Fluid Mech. 254* (1993), 323–344.

[60] Pan, T., Joseph, D., Bai, R., Glowinski, R., and Sarin, V. Fluidization of 1204 spheres: simulation and experiment. *Journal of Fluid Mechanics 451* (2002), 169–191.

[61] Peskin, C., and McQueen, D. A three-dimensional computational method for blood flow in the heart: I. Immersed elastic fibers in a viscous incompressible fluid. *J. Comp. Physics 81* (1989), 372–405.

[62] Pironneau, O. On the transport-diffusion algorithm and its applications to the Navier-Stokes equations. *Numer. Math. 38* (1982), 309–332.

[63] Roshko, Anatol. . *Phys. Fluids 10* (1967), 5181–5183.

[64] Sakamoto, H., and Hanu, H. A study on vortex shedding from spheres in a uniform flow. *J. Fluids Engng. 112* (1990), 386–392.

[65] Sanders, S. *Slurry Transport.* Syncrude Research Centre, 2002. www.syncrude.com/research/.

[66] Sanders, S. *Underground Injection Control Program.What is Hydraulic Fracturing?* U.S. Environmental Protection Agency (EPA), 2002. www.epa.gov/safewater/uic/cbmstudy/hfracdef.html.

[67] Singh, P., and Joseph, D. D. Sedimentation of a sphere near a vertical wall in an Oldroyd-b fluid. *J. Non-Newtonian Fluid Mech. 94* (2000).

[68] Taneda, S. Experimental investigation of th wake behind a sphere at low Reynolds numbers. *J. Phys. Soc. Jpn. 11* (1956), 1104–1108.

[69] Taneda, S. Experimental investigation of the wakes behind cylinders and plates at low Reynolds numbers. *J. Phys. Soc. Jpn. 11* (1956), 302–307.

[70] Temam, R. *Navier-Stokes equations.* North-Holland, Amsterdam, New York., 1984.

[71] Timmermans, L. *Analysis of spectral element methods with applications to incompressible flow.* PhD thesis, Eindhoven University of Technology, 1994.

[72] Tomboulides, A., Orszag, S., and Karniadakis, G. Direct and large-eddy simulation of axisymmetric wakes. In *Lecture Notes in Computer Science* (1993), AIAA. Paper 93-0546.

[73] Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S., and Jan, Y.-J. A front-tracking method for viscous incompressible multi-fluid flows. *J. Comp. Physics 169* (2001), 708.

[74] Unverdi, S., and Tryggvason, G. A front-tracking method for viscous incompressible multi-fluid flows. *Journal of Comp. Physics 100* (1992), 25–37.

[75] van Kan, J. A second order accurate pressure-correction scheme for viscous incompressible flow. *J. Sci. Stat. Comp. 7* (1986), 870–891.

[76] Wilkinson, B., and Allen, M. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers.* Prentice Hall, 1999.

[77] Zhang, T., and Prosperitti, A. Averaged equations for inviscid disperse two-phase flow. *Journal of Fluid Mechanics 267* (1994), 185–200.

Appendix A. Numerical integration

A.1 Adapted Gaussian quadrature

A.1.1 One dimensional Gauss quadrature

Gaussian quadrature seeks to obtain the best numerical estimate of an integral by picking optimal abscissas at which to evaluate the function $f(t)$ [42]. The fundamental theorem of Gaussian quadrature states that the optimal abscissas of the m-point Gaussian quadrature formulas are precisely the roots of the orthogonal polynomial for the same interval and weighting function. Gaussian quadrature is optimal because it fits all polynomials up to degree exactly. The simplest form of Gaussian Integration is based on the use of an optimally chosen polynomial to approximate the integrand $f(t)$ over the interval $[-1, +1]$. It can be shown that the best estimate of the integral is then:

$$\int_{-1}^{1} f(t)dt = \sum_{i=1}^{n} w_i f(t_i) \tag{A.1}$$

where t_i is a designated evaluation point, and w_i is the weight of that point in the sum. If the number of points at which the function $f(t)$ is evaluated is n, the resulting value of the integral is of the same accuracy as a simple polynomial method.

The Gauss-Legendre integration formula given before evaluates the integral on the interval for t of $[-1, +1]$. In most cases the integral needs to be evaluated on a more general interval, say $[\alpha, \beta]$. This can be achieved by mapping the interval $[\alpha, \beta]$ onto the [-1,+1] interval using an appropriate transformation.

A.1.2 Multi-dimensional Gauss quadrature

Multi-dimension quadrature rules can be obtained from products of one-dimensional quadrature rules. It is convenient to map the region of integration to a convenient unit area or volume. This can be done using an isoparametric transformation. The

space used to define the element shape functions is the best choice for the region of integration. The essential idea underlying this centres on the mapping of the simple geometric shape in the local coordinate system (ξ, η, ζ) into distorted shapes in the global Cartesian coordinate system (x, y, z). The mapping from local to global coordinates will take the form

$$x = \sum_{i=1}^{N_v} x_i \, \phi_i(\xi, \eta, \zeta) \tag{A.2}$$

$$y = \sum_{i=1}^{N_v} y_i \, \phi_i(\xi, \eta, \zeta) \tag{A.3}$$

$$z = \sum_{i=1}^{N_v} z_i \, \phi_i(\xi, \eta, \zeta) \tag{A.4}$$

where N_v is the number of points defining the geometry of the element, ϕ_i is the ith basis function and x_i, y_i, z_i are the Cartesian coordinates of the nodal points of the element. There is a one-to-one correspondence between the nodes in the standard element and the element in the global coordinate system. Due to the transformation equation, the volume integral for a given element can be written as:

$$I = \int_\Omega f(x, y, z) dx dy dz = \int_\Omega f(\xi, \eta, \zeta) |\mathbf{J}| d\xi d\eta d\zeta \tag{A.5}$$

where $\mathbf{J}$ is the transformation Jacobian given by:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{pmatrix} \tag{A.6}$$

and $|\mathbf{J}|$ is the determinant of $\mathbf{J}$.

The integral equation can be numerically integrated by using Gaussian quadrature. This yields

$$I = \sum_{i=1}^{N_v} \sum_{j=1}^{N_v} \sum_{k=1}^{N_v} w_{i,j,k} f(\xi_i, \eta_j, \zeta_k) |\mathbf{J}| \tag{A.7}$$

where $w_{i,j,k}$ is the weight corresponding to the local coordinate point (ξ_i, η_j, ζ_k).

Derivatives are easily converted from one coordinate system to another by means of the chain rule of partial differentiation. For example:

$$\frac{\partial f(x, y, z)}{\partial x} = \frac{\partial f(\xi, \eta, \zeta)}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial f(\xi, \eta, \zeta)}{\partial \eta} \frac{\partial \eta}{\partial x} + \frac{\partial f(\xi, \eta, \zeta)}{\partial \zeta} \frac{\partial \zeta}{\partial x} \tag{A.8}$$